

504p074/w000

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-299857

(P 2 0 0 0 - 2 9 9 8 5 7 A)

(43) 公開日 平成12年10月24日 (2000. 10. 24)

(51) Int. Cl. ⁷	識別記号	F I	テマコード (参考)
H04N 7/24		H04N 7/13	Z 5C053
5/91		5/91	Z 5C059
5/92		5/92	H

審査請求 未請求 請求項の数129 O L (全100頁)

(21) 出願番号	特願平11-343157	(71) 出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22) 出願日	平成11年12月2日 (1999. 12. 2)	(72) 発明者	北村 卓也 東京都品川区北品川6丁目7番35号 ソニー株式会社内
(31) 優先権主張番号	特願平11-31944	(74) 代理人	100082131 弁理士 稲本 義雄
(32) 優先日	平成11年2月9日 (1999. 2. 9)		
(33) 優先権主張国	日本 (J P)		

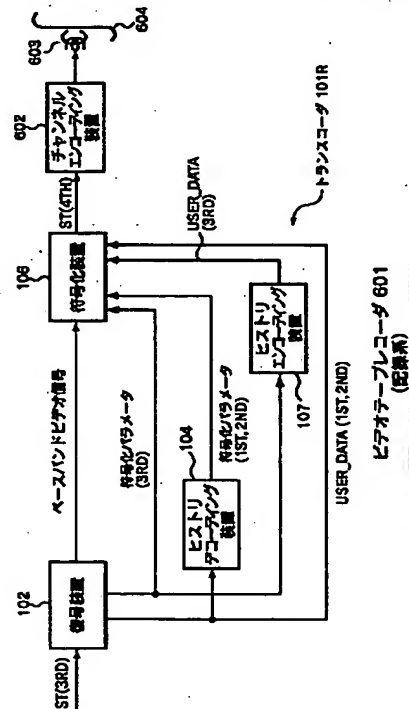
最終頁に続く

(54) 【発明の名称】 符号化ストリーム記録装置および方法、符号化ストリーム再生装置および方法、トランスコーディングシステム、変換装置および方法、ビデオ符号化装置および方法、ストリーム処理装置および方法

(57) 【要約】

【課題】 トランスコーディングされたデータをビデオテープに記録する場合にも、画質の劣化を抑制できるようにする。

【解決手段】 符号化装置106は、復号装置102より入力されたベースバンドビデオ信号に対して、符号化処理を行う。このとき、復号装置102、またはヒストリデコーディング装置104より供給される第1世代乃至第3世代の符号化パラメータが利用できる場合、それを利用して符号化処理が行われる。符号化装置106はまた、第1世代乃至第3世代の過去の符号化パラメータをピクチャ層のuser_dataに記録し、第4世代の符号化ストリームSTに多重化して、チャンネルエンコーディング装置602に出力する。チャンネルエンコーディング装置602は、入力された符号化ストリームをチャンネルエンコードし、記録ヘッド603を介して、磁気テープ604に記録させる。



【特許請求の範囲】

【請求項 1】 入力された符号化ストリームを記録媒体に記録する符号化ストリーム記録装置において、入力された第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出手段と、前記検出手段の検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化手段と、前記符号化手段により生成された前記第 2 の符号化ストリームと、前記検出手段により検出された前記過去の符号化処理における符号化履歴とを、前記記録媒体に記録する記録手段とを備えることを特徴とする符号化ストリーム記録装置。

【請求項 2】 前記符号化手段は、MPEG方式で符号化を行うことを特徴とする請求項 1 に記載の符号化ストリーム記録装置。

【請求項 3】 前記記録手段は、前記符号化履歴を、前記MPEG方式の前記第 2 の符号化ストリームのuser_dataとして多重化して記録することを特徴とする請求項 2 に記載の符号化ストリーム記録装置。

【請求項 4】 前記記録手段は、前記符号化履歴を、前記第 2 の符号化ストリームに較べて、同期コードからより離れた位置に多重化して記録することを特徴とする請求項 1 に記載の符号化ストリーム記録装置。

【請求項 5】 入力された符号化ストリームを記録媒体に記録する符号化ストリーム記録装置の符号化ストリーム記録方法において、入力された第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、前記検出ステップの処理における検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化ステップと、前記符号化ステップの処理により生成された前記第 2 の符号化ストリームと、前記検出ステップの処理により検出された前記過去の符号化処理における符号化履歴とを、前記記録媒体に記録する記録ステップとを含むことを特徴とする符号化ストリーム記録方法。

【請求項 6】 入力された符号化ストリームを記録媒体に記録する符号化ストリーム記録装置を制御するプログラムにおいて、入力された第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、前記検出ステップの処理における検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化ステップと、前記符号化ステップの処理により生成された前記第 2 の符号化ストリームと、前記検出ステップの処理により検出された前記過去の符号化処理における符号化履歴とを、前記記録媒体に記録する記録ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラム

が記録されている記録媒体。

【請求項 7】 記録媒体に記録されている符号化ストリームを再生する符号化ストリーム再生装置において、前記記録媒体に記録されている第 1 の符号化ストリームを再生する再生手段と、前記再生手段により再生された前記第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出手段と、前記検出手段の検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化手段と、前記符号化手段により生成された前記第 2 の符号化ストリームと、前記検出手段により検出された前記過去の符号化処理における符号化履歴とを、多重化して出力する出力手段とを備えることを特徴とする符号化ストリーム再生装置。

【請求項 8】 前記符号化手段は、前記第 1 の符号化ストリームをMPEG方式で符号化することを特徴とする請求項 7 に記載の符号化ストリーム再生装置。

【請求項 9】 前記検出手段は、前記符号化履歴を、前記MPEG方式の前記第 1 の符号化ストリームのuser_dataから検出することを特徴とする請求項 8 に記載の符号化ストリーム再生装置。

【請求項 10】 前記検出手段は、前記符号化履歴を、前記第 1 の符号化ストリームに較べて、同期コードからより離れた位置から検出することを特徴とする請求項 7 に記載の符号化ストリーム再生装置。

【請求項 11】 記録媒体に記録されている符号化ストリームを再生する符号化ストリーム再生装置の符号化ストリーム再生方法において、前記記録媒体に記録されている第 1 の符号化ストリームを再生する再生ステップと、前記再生ステップの処理により再生された前記第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、前記検出ステップの処理における検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化ステップと、前記符号化ステップの処理により生成された前記第 2 の符号化ストリームと、前記検出ステップの処理により検出された前記過去の符号化処理における符号化履歴とを、多重化して出力する出力ステップとを含むことを特徴とする符号化ストリーム再生方法。

【請求項 12】 記録媒体に記録されている符号化ストリームを再生する符号化ストリーム再生装置を制御するプログラムにおいて、前記記録媒体に記録されている第 1 の符号化ストリームを再生する再生ステップと、前記再生ステップの処理により再生された前記第 1 の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、

前記検出ステップの処理における検出結果を利用して、前記第 1 の符号化ストリームを符号化して第 2 の符号化ストリームを生成する符号化ステップと、

前記符号化ステップの処理により生成された前記第 2 の符号化ストリームと、前記検出ステップの処理により検出された前記過去の符号化処理における符号化履歴とを、多重化して出力する出力ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 13】 入力符号化ビデオストリームのビットレートまたは GOP 構造を変換するトランスコーディングシステムにおいて、

前記入力符号化ビデオストリームを復号してビデオデータを生成するとともに、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、前記過去の符号化パラメータを受け取り、前記過去の符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 14】 前記符号化手段は、前記現在の符号化処理において、前記入力ビデオデータに含まれる参照ピクチャに割り当てられた現在のピクチャタイプで、前記参照ピクチャをエンコードし、

前記制御手段は、過去の符号化処理において割り当てられたピクチャタイプと同一のピクチャタイプに前記参照ピクチャが符号化されたか否かを判定し、その判定結果に基づいて、前記現在の符号化処理を制御することを特徴とする請求項 13 に記載のトランスコーディングシステム。

【請求項 15】 前記符号化手段は、前記ビデオデータに含まれる参照ピクチャを、前記現在の符号化処理において、前記参照ピクチャに割り当てられた現在のピクチャタイプで符号化し、

前記制御手段は、前記現在の符号化処理において、前記参照ピクチャに割り当てられた過去のピクチャタイプを、前記過去の符号化パラメータを参照することで検出し、前記現在のピクチャタイプと前記過去のピクチャタイプに基づいて前記現在の符号化処理を制御することを特徴とする請求項 13 に記載のトランスコーディングシステム。

【請求項 16】 前記制御手段は、前記判定に基づいて、前記過去の符号化パラメータから最適な符号化パラメータを選択し、前記選択された最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御することを特徴とする請求項 14 に記載のトランスコーディングシステム。

【請求項 17】 前記制御手段は、前記参照ピクチャ

を、前記過去の符号化処理において発生された過去の符号化パラメータの 1 つを使用して符号化することを特徴とする請求項 14 に記載のトランスコーディングシステム。

【請求項 18】 前記過去の符号化パラメータは、前記過去の符号化処理において生成された動きベクトル情報を含み、

前記符号化手段は、前記現在の符号化処理において、前記参照ピクチャの動きベクトル情報を検出するための動きベクトル検出手段を含むことを特徴とする請求項 16 に記載のトランスコーディングシステム。

【請求項 19】 前記制御手段は、前記判断の結果に基づいて、前記動きベクトル検出手段の動作を制御することを特徴とする請求項 18 に記載のトランスコーディングシステム。

【請求項 20】 前記制御手段は、前記動きベクトル検出手段における新たな動きベクトル情報の計算の代わりに、前記過去の符号化パラメータに含まれる、前記動きベクトル情報を再使用することを特徴とする請求項 19 に記載のトランスコーディングシステム。

【請求項 21】 前記参照ピクチャが過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、前記過去の符号化処理に含まれる前記動きベクトル情報を再使用することを特徴とする請求項 19 に記載のトランスコーディングシステム。

【請求項 22】 前記参照ピクチャが過去の符号化処理において、前記割り当てられたピクチャタイプで符号化されていない場合、前記制御手段は、新たな動きベクトル情報が、前記動きベクトル検出手段により検出されたように前記動きベクトル検出手段を制御することを特徴とする請求項 21 に記載のトランスコーディングシステム。

【請求項 23】 前記制御手段は、前記過去の符号化パラメータから、前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の前記現在の符号化処理を制御することを特徴とする請求項 13 に記載のトランスコーディングシステム。

【請求項 24】 前記符号化手段は、前記現在の符号化処理において、参照ピクチャに割り当てられた現在のピクチャタイプで前記入力ビデオデータに含まれる前記参照ピクチャを符号化し、

前記制御手段は、前記参照ピクチャが過去の符号化処理において割り当てられたピクチャタイプと同一のピクチャタイプに符号化されたか否かを判定し、その判定結果に基づいて、前記最適な符号化パラメータを選択することを特徴とする請求項 13 に記載のトランスコーディングシステム。

【請求項 25】 前記過去の符号化パラメータは、フレ

ーム予測モードまたはフィールド予測モードを表す予測モード情報を含み、

前記制御手段は、前記予測モード情報に対応して、前記現在の符号化処理を制御することを特徴とする請求項23に記載のトランスコーディングシステム。

【請求項26】 前記参照ピクチャが、前記過去の符号化処理において、前記過去のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、新たな予測モード情報の計算に代えて、前記過去の符号化パラメータに含まれる前記予測モード情報を再使用する

ことを特徴とする請求項23に記載のトランスコーディングシステム。

【請求項27】 前記符号化パラメータは、イントラ予測、前方予測、後方予測、または双方向予測を示す予測タイプ情報を含み、

前記制御手段は、前記予測タイプ情報に基づいて、前記現在の符号化処理を制御することを特徴とする請求項23に記載のトランスコーディングシステム。

【請求項28】 前記参照ピクチャが、過去の符号化処理において、前記現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、新たな予測タイプ情報の計算に代えて、前記過去の符号化

パラメータに含まれる前記予測タイプ情報を再使用することを特徴とする請求項27に記載のトランスコーディングシステム。

【請求項29】 前記符号化パラメータは、フレームDCTモードまたはフィールドDCTモードを表すDCTモード情報を含み、

前記制御手段は、前記DCTモード情報に基づいて、前記現在の符号化処理を制御することを特徴とする請求項23に記載のトランスコーディングシステム。

【請求項30】 前記参照ピクチャが過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、新たなDCTモード情報の計算に代えて、前記過去の符号化パラメータに含まれる、前記DCTモード情報を再使用することを特徴とする請求項29に記載のトランスコーディングシステム。

【請求項31】 前記制御手段は、前記符号化手段の現在の符号化処理に対応する現在の符号化パラメータを生成することを特徴とする請求項13に記載のトランスコーディングシステム。

【請求項32】 前記制御手段は、前記現在の符号化パラメータと過去の符号化パラメータから、前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の前記現在の符号化処理を制御することを特徴とする請求項29に記載のトランスコーディングシステム。

【請求項33】 前記符号化手段は、前記入力ビデオデータに含まれる参照ピクチャを、前記現在の符号化処理

において、前記参照ピクチャに割り当てられた現在のピクチャタイプで符号化し、

前記制御手段は、前記参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて最適な符号化パラメータを選択することを特徴とする請求項32に記載のトランスコーディングシステム。

【請求項34】 前記過去の符号化パラメータは、前記過去の符号化処理において生成された量子化情報を含み、

前記符号化手段は、前記現在の符号化処理において、前記参照ピクチャを量子化する量子化手段を含むことを特徴とする請求項33に記載のトランスコーディングシステム。

【請求項35】 前記制御手段は、前記符号化ビデオストリームを蓄積する伝送バッファの残量を表すバッファ情報を受け取り、前記バッファ情報に基づいて、前記伝送バッファのオーバーフローおよびアンダーフローを阻止するように、前記量子化手段を制御することを特徴とする請求項34に記載のトランスコーディングシステム。

【請求項36】 前記制御手段は、前記バッファ情報から抽出した量子化ステップサイズ、および前記過去の符号化パラメータに含まれる前記量子化情報から抽出した量子化ステップサイズに基づいて、前記量子化手段を制御することを特徴とする請求項35に記載のトランスコーディングシステム。

【請求項37】 前記制御手段は、前記バッファ情報に対応する量子化ステップサイズと、前記量子化情報に対応する量子化ステップサイズから選択した最も大きい量子化ステップサイズを使用して、前記量子化手段を制御することを特徴とする請求項36に記載のトランスコーディングシステム。

【請求項38】 前記制御手段は、前記符号化手段が前記過去の符号化パラメータを前記符号化ビデオストリームに記述するように制御することを特徴とする請求項13に記載のトランスコーディングシステム。

【請求項39】 前記符号化手段は、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有するMPEG標準に従ったMP EGビットストリームを、前記符号化ビデオストリームを処理して生成する処理手段を有することを特徴とする請求項13に記載のトランスコーディングシステム。

【請求項40】 前記制御手段は、前記符号化手段の前記現在の符号化処理に対応する現在の符号化パラメータを生成し、

前記処理手段は、前記現在の符号化パラメータを前記ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤに記述し、前記過去の符号化パラメータを前記ピ

クチャレイヤのユーザデータエリアに記述することを特徴とする請求項 39 に記載のトランスコーディングシステム。

【請求項 41】 前記処理手段は、前記ユーザデータエリアに、前記過去の符号化パラメータを記述するために、過去の符号化パラメータを含むヒストリストリームを生成することを特徴とする請求項 40 に記載のトランスコーディングシステム。

【請求項 42】 前記処理手段は、MPEG標準に規定されているスタートコードのエミレーションを防止するために、ヒストリストリームにマーカビットを挿入し、前記マーカビットが挿入されたヒストリストリームを前記ピクチャレイヤのユーザデータエリアに記述することを特徴とする請求項 41 に記載のトランスコーディングシステム。

【請求項 43】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、前記入力符号化ビデオストリームを復号し、復号ビデオデータを生成する生成ステップと、前記入力符号化ビデオストリームから過去の符号化処理において生成された符号化パラメータを抽出する抽出ステップと、現在の符号化処理により、符号化ビデオストリームを生成するように、前記ビデオデータを符号化する符号化ステップと、生成された前記過去の符号化パラメータを受け取る受け取りステップと、前記過去の符号化パラメータに基づいて、前記符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする変換方法。

【請求項 44】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出し、前記過去の符号化パラメータをヒストリ情報として出力する復号手段と、前記復号されたビデオデータを符号化し、符号化ビデオストリームを現在の符号化処理として生成する符号化手段と、

過去の符号化パラメータを含む、前記ヒストリ情報を受け取り、前記過去の符号化パラメータを選択的に使用することで、前記現在の符号化処理が最適化されるように、前記ヒストリ情報に基づいて、前記符号化手段の前記現在の符号化処理を制御する制御手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 45】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、入力符号化ビデオストリームを復号し、復号ビデオデー

タを生成する復号ステップと、

前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

前記過去の符号化パラメータをヒストリ情報として出力する出力ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する生成ステップと、

過去の符号化パラメータを含む前記ヒストリ情報を受け取る受け取りステップと、

前記現在の符号化処理が、前記過去の符号化パラメータを選択的に使用することで最適化されるように、前記ヒストリ情報に基づいて、前記符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする変換方法。

【請求項 46】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

前記過去の符号化パラメータを受け取り、前記過去の符号化パラメータから前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 47】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する生成ステップと、

前記過去の符号化パラメータを受け取る受け取りステップと、

前記過去の符号化パラメータから前記現在の符号化処理に対応する最適な符号化パラメータを選択する選択ステップと、

前記最適な符号化パラメータに基づいて、前記符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする変換方法。

【請求項 4 8】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、
 入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、
 割り当てられたピクチャタイプで、前記符号化ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、
 過去の符号化処理において生成された過去の符号化パラメータを受け取り、前記割り当てられたピクチャタイプに従って、前記過去の符号化パラメータから最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 4 9】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、
 前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する生成ステップと、
 前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、
 割り当てられたピクチャタイプで、前記符号化ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、
 過去の符号化処理において生成された過去の符号化パラメータを受け取る受け取りステップと、
 前記割り当てられたピクチャタイプに従って、前記過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、
 前記最適な符号化パラメータに基づいて、前記符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする変換方法。

【請求項 5 0】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、
 入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、
 割り当てられたピクチャタイプで、前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、
 前記参照ピクチャが過去の符号化処理において、前記割り当てられたピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 5 1】 入力符号化ビデオストリームのビット

レートまたはGOP構造を変換する変換方法において、
 前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ステップと、
 前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、
 割り当てられたピクチャタイプで、前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、
 前記参照ピクチャが前記過去の符号化処理において、前記割り当てられたピクチャタイプで符号化されたか否かを判定する判定ステップと、
 前記判定の結果に基づいて、前記符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とする変換方法。

【請求項 5 2】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、
 前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、
 現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、
 前記符号化ビデオストリーム、前記現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームを生成するストリーム生成手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 5 3】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、
 前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、
 前記入力符号化ビデオストリームから、過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、
 現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、
 前記符号化ビデオストリーム、前記現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームを生成するMPEGストリーム生成ステップとを含むことを特徴とする変換方法。

【請求項 5 4】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、
 前記入力符号化ビデオストリームを復号して、復号ビデオ

10

20

30

40

50

オデータを生成し、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームを生成するストリーム生成手段とを備え、

前記各レイヤは、前記現在の符号化処理において生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理において生成された前記過去の符号化パラメータを含むことを特徴とするトランスコーディングシステム。

【請求項55】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、前記入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、

シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームを生成するMPEGビットストリーム生成ステップとを含み、

前記各レイヤは、前記現在の符号化処理において生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理において生成された過去の符号化パラメータを含むことを特徴とする変換方法。

【請求項56】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

前記過去の符号化パラメータを参照することにより、現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述手段と、

前記過去の符号化パラメータが記述されている前記符号化ビデオストリームを出力する出力手段とを備えることを特徴とするトランスコーディングシステム。

【請求項57】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

前記過去の符号化パラメータを参照することにより、現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

10 前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述ステップと、

前記過去のパラメータが記述されている前記符号化ビデオストリームを出力する出力ステップとを含むことを特徴とする変換方法。

【請求項58】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成手段と、

20 前記入力符号化ビデオストリームから、過去の符号化処理により生成された以前の符号化パラメータを含むヒストリ情報を抽出する抽出手段と、

前記ヒストリ情報を参照することにより、現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成手段と、

前記ヒストリ情報が、将来の符号化処理において利用できるように、前記符号化ストリームに記述する記述手段とを備えることを特徴とするトランスコーディングシステム。

【請求項59】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、

入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、

前記入力符号化ビデオストリームから、過去の符号化処理により生成された前の符号化パラメータを含むヒストリ情報を抽出する抽出ステップと、

前記ヒストリ情報を参照することにより、現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、

前記ヒストリ情報が、将来の符号化処理において利用できるように前記符号化ストリームに前記ヒストリ情報を記述する記述ステップとを含むことを特徴とする変換方法。

【請求項60】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換するトランスコーディングシステムにおいて、

直近の符号化処理の符号化パラメータに基づいて、前記入力符号化ビデオストリームを復号してベースバンドビ

デオデータを生成し、過去の符号化処理の過去の符号化パラメータを抽出し、前記直近および過去の符号化処理の前記符号化パラメータを前記ベースバンドビデオデータに多重化する復号手段と、

新たな符号化ビデオストリームのビットレートまたはGOP構造が、前記入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、前記直近および過去の符号化処理の符号化パラメータに基づいて、前記ベースバンドビデオデータを符号化し、前記新たな符号化ビデオストリームを生成する符号化手段とを備えることを特徴とするトランスコーディングシステム。

【請求項 6 1】 入力符号化ビデオストリームのビットレートまたはGOP構造を変換する変換方法において、直近の符号化処理の符号化パラメータに基づいて、前記入力符号化ビデオストリームを符号化して、ベースバンドビデオデータを生成するベースバンドビデオデータ生成ステップと、

過去の符号化処理の過去の符号化パラメータを抽出する抽出ステップと、

前記直近および過去の符号化処理の前記符号化パラメータを前記ベースバンドビデオデータに多重化する多重化ステップと、

新たな符号化ビデオストリームのビットレートまたはGOP構造が、前記入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、前記直近および過去の符号化処理の符号化パラメータに基づいて、前記ベースバンドビデオデータを符号化して、前記新たな符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップとを含むことを特徴とする変換方法。

【請求項 6 2】 入力ビデオデータを符号化するビデオ符号化装置において、

現在の符号化処理として、前記入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化手段と、

過去の符号化処理により生成された過去の符号化パラメータを受け取り、前記過去の符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするビデオ符号化装置。

【請求項 6 3】 前記符号化手段は、前記現在の符号化処理において、前記入力ビデオデータに含まれる参照ピクチャを、前記参照ピクチャに割り当てられた現在のピクチャタイプで符号化し、

前記制御手段は、前記参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、前記現在の符号化処理を前記判定結果に基づいて制御することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 6 4】 前記符号化手段は、前記現在の符号化処理において、前記入力ビデオデータに含まれる参照ピクチャを、割り当てられた現在のピクチャタイプで符号

化し、

前記制御手段は、現在の符号化処理で、前記参照ピクチャに割り当てられた過去のピクチャタイプを、前記過去の符号化パラメータを参照することにより検出し、前記現在のピクチャタイプおよび過去のピクチャタイプに基づいて、前記現在の符号化処理を制御することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 6 5】 前記制御手段は、前記判定結果に従って、前記過去の符号化パラメータから最適な符号化パラメータを選択し、前記選択した最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御することを特徴とする請求項 6 3 に記載のビデオ符号化装置。

【請求項 6 6】 前記制御手段は、1つの前記過去の符号化処理で生成された前記過去の符号化パラメータを使用して前記参照ピクチャを符号化することを特徴とする請求項 6 3 に記載のビデオ符号化装置。

【請求項 6 7】 前記過去の符号化パラメータは、前記過去の符号化処理において生成された動きベクトルを含み、

前記符号化手段は、前記現在の符号化処理において、前記参照ピクチャの動きベクトル情報を検出する動きベクトル検出手段を含むことを特徴とする請求項 6 5 に記載のビデオ符号化装置。

【請求項 6 8】 前記制御手段は、前記判定結果に基づいて前記動きベクトル検出手段の処理を制御することを特徴とする請求項 6 7 に記載のビデオ符号化装置。

【請求項 6 9】 前記制御手段は、前記動きベクトル検出手段において、新たな動きベクトル情報を計算する代わりに、前記過去の符号化パラメータに含まれる前記動きベクトル情報を再使用することを特徴とする請求項 6 8 に記載のビデオ符号化装置。

【請求項 7 0】 前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、前記過去の符号化処理に含まれる動きベクトル情報を再使用することを特徴とする請求項 6 8 に記載のビデオ符号化装置。

【請求項 7 1】 前記制御手段は、前記参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプで符号化されていない場合、新たな動きベクトル情報が、前記動きベクトル検出手段で検出されたように、前記動きベクトル検出手段を制御することを特徴とする請求項 7 0 に記載のビデオ符号化装置。

【請求項 7 2】 前記制御手段は、前記過去の符号化パラメータから、前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 7 3】 前記符号化手段は、前記現在の符号化処理において、前記入力ビデオデータに含まれる参照ピクチャを、割り当てられた現在のピクチャタイプで符号化し、

前記制御手段は、前記参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、前記判定結果に基づいて、前記最適な符号化パラメータを選択することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 7 4】 前記過去の符号化パラメータは、フレーム予測モードまたはフィールド予測モードを表す予測モード情報を含み、

前記制御手段は、前記予測モード情報に従って、前記現在の符号化処理を制御することを特徴とする請求項 7 2 に記載のビデオ符号化装置。

【請求項 7 5】 前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、新たな予測モード情報を計算する代わりに、前記過去の符号化パラメータに含まれる前記予測モード情報を再使用することを特徴とする請求項 7 4 に記載のビデオ符号化装置。

【請求項 7 6】 前記符号化パラメータは、イントラ予測、前方予測、後方予測、または両方向予測を表す予測タイプ情報を含み、

前記制御手段は、前記予測タイプ情報に基づいて、前記現在の符号化処理を制御することを特徴とする請求項 7 2 に記載のビデオ符号化装置。

【請求項 7 7】 前記制御手段は、前記参照ピクチャが、前記現在のピクチャタイプと同一のピクチャタイプで過去の符号化処理において符号化されている場合、新たな予測タイプ情報を計算する代わりに、前記過去の符号化パラメータに含まれる前記予測タイプ情報を再使用することを特徴とする請求項 7 6 に記載のビデオ符号化装置。

【請求項 7 8】 前記符号化パラメータは、フレーム DCT モードまたはフィールド DCT モードを表す DCT モード情報を含み、

前記制御手段は、前記 DCT モード情報に基づいて、前記現在の符号化処理を制御することを特徴とする請求項 7 2 に記載のビデオ符号化装置。

【請求項 7 9】 前記参照ピクチャが、過去の符号化処理において、前記現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、前記制御手段は、新たな DCT モード情報を計算する代わりに、前記過去の符号化パラメータに含まれる前記 DCT モード情報を再使用することを特徴とする請求項 7 8 に記載のビデオ符号化装置。

【請求項 8 0】 前記制御手段は、前記符号化手段の前記現在の符号化処理に対応する現在の符号化パラメータ

を生成することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 8 1】 前記制御手段は、前記現在の符号化パラメータと前記過去の符号化パラメータから、前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の前記現在の符号化処理を制御することを特徴とする請求項 8 0 に記載のビデオ符号化装置。

【請求項 8 2】 前記符号化手段は、前記現在の符号化処理において、前記入力ビデオデータに含まれる前記参照ピクチャを、前記参照ピクチャに割り当てられた現在のピクチャタイプで符号化し、

前記制御手段は、前記参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて、前記最適な符号化パラメータを選択することを特徴とする請求項 8 1 に記載のビデオ符号化装置。

【請求項 8 3】 前記過去の符号化パラメータは、前記過去の符号化処理で生成された量子化情報を含み、

前記符号化手段は、前記現在の符号化処理において前記参照ピクチャを量子化する量子化手段を含むことを特徴とする請求項 8 2 に記載のビデオ符号化装置。

【請求項 8 4】 前記制御手段は、前記符号化ビデオストリームを蓄積する伝送バッファの残量を表すバッファ情報を受け取り、前記伝送バッファのオーバーフローおよびアンダーフローの発生を抑制するように、前記バッファ情報に基づいて前記量子化手段を制御することを特徴とする請求項 8 3 に記載のビデオ符号化装置。

【請求項 8 5】 前記制御手段は、前記バッファ情報から生成された量子化ステップサイズ、および前記過去の符号化パラメータに含まれる前記量子化情報から生成された量子化ステップサイズに基づいて、前記量子化手段を制御することを特徴とする請求項 8 4 に記載のビデオ符号化装置。

【請求項 8 6】 前記制御手段は、前記バッファ情報に対応する前記量子化ステップサイズ、および前記量子化情報に対応する前記量子化ステップサイズの中から選択された最大の量子化ステップサイズを利用して、前記量子化手段を制御することを特徴とする請求項 8 5 に記載のビデオ符号化装置。

【請求項 8 7】 前記制御手段は、前記符号化手段が前記過去の符号化パラメータを前記符号化ビデオストリームに記述するように前記符号化手段を制御することを特徴とする請求項 6 2 に記載のビデオ符号化装置。

【請求項 8 8】 前記符号化手段は、シーケンスレイヤ、GOP レイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有する MPEG 標準に基づく MP EG ビットストリームを生成するように、前記符号化ビデオストリームを処理する処理手段を含むことを特徴とす

る請求項 6 2 に記載のビデオ符号化装置。

【請求項 8 9】 前記制御手段は、前記符号化手段の現在の符号化処理に対応する現在の符号化パラメータを発生し、

前記符号化手段は、前記ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤに前記現在の符号化パラメータを記述し、前記過去の符号化パラメータを前記ピクチャレイヤのユーザデータエリアに記述することを特徴とする請求項 8 8 に記載のビデオ符号化装置。

【請求項 9 0】 前記符号化手段は、前記ユーザデータエリアに、前記過去の符号化パラメータを記述するために、前記過去の符号化パラメータを含むヒストリストリームを発生することを特徴とする請求項 8 9 に記載のビデオ符号化装置。

【請求項 9 1】 前記符号化手段は、MPEG 標準に指定されているスタートコードのエミレーションの発生を防止するために、前記ヒストリストリームにマーカビットを挿入し、前記ピクチャレイヤの前記ユーザデータエリアに前記マーカビットが挿入された前記ヒストリストリームを記述することを特徴とする請求項 9 0 に記載のビデオ符号化装置。

【請求項 9 2】 入力ビデオデータを符号化するビデオ符号化装置において、

現在の符号化処理として、前記入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化手段と、

過去の符号化処理で生成された過去の符号化パラメータを含むヒストリ情報を受け取り、前記現在の符号化処理が前記過去の符号化パラメータを選択的に使用することにより最適化されるように、前記ヒストリ情報に基づいて前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするビデオ符号化装置。

【請求項 9 3】 入力ビデオデータを符号化するビデオ符号化方法において、現在の符号化処理として、前記入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化ステップと、

過去の符号化処理で発生された過去の符号化パラメータを含むヒストリ情報を受け取り、前記現在の符号化処理が前記過去の符号化パラメータを選択的に使用することで最適化されるように、前記ヒストリ情報に基づいて前記符号化ステップでの処理の現在の符号化処理を制御する制御ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 9 4】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化装置において、現在の符号化処理として、前記ビデオデータを符号化して、前記符号化ビデオストリームを生成する符号化手段と、

過去の符号化処理で発生された過去の符号化パラメータを受け取り、前記過去の符号化パラメータから前記現在

の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするビデオ符号化装置。

【請求項 9 5】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化方法において、現在の符号化処理として、前記ビデオデータを符号化して、前記符号化ビデオストリームを生成する符号化ステップ、

過去の符号化処理で発生された過去の符号化パラメータを受け取り、前記過去の符号化パラメータから前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 9 6】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化装置において、割り当てられたピクチャタイプで、前記ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、

過去の符号化処理で生成された過去の符号化パラメータを受け取り、前記割り当てられたピクチャタイプに対応して、前記過去の符号化パラメータから最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とするビデオ符号化装置。

【請求項 9 7】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化方法において、割り当てられたピクチャタイプで、前記ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、

過去の符号化処理で生成された過去の符号化パラメータを受け取り、前記割り当てられたピクチャタイプに従って、前記過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、

前記最適な符号化パラメータに基づいて、前記符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 9 8】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化装置において、割り当てられたピクチャタイプで、前記ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、

過去の符号化処理において、前記割り当てられたピクチャタイプで前記参照ピクチャが符号化されたかどうかを判定し、前記判定結果に基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするビデオ符号化装置。

【請求項 9 9】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化方法において、割り当てられたピクチャタイプで、前記ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、前記参照ピクチャが、過去の符号化処理において、前記

割り当てられたピクチャタイプで符号化されたか否かを判定する判定ステップと、

前記判定ステップでの前記判定結果に基づいて、前記符号化ステップでの現在の符号化処理を制御する制御ステップとを含むこと特徴とするビデオ符号化方法。

【請求項 100】 入力ビデオデータを符号化するビデオ符号化装置において、

現在の符号化処理として、前記入力ビデオデータを符号化し、符号化ビデオデータを生成する符号化手段と、

前記符号化ビデオデータ、前記現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含む MPEG ビットストリームを生成するストリーム生成手段とを備えること特徴とするビデオ符号化装置。

【請求項 101】 入力ビデオデータを符号化するビデオ符号化方法において、

現在の符号化処理として、符号化ビデオデータを発生する符号化ビデオデータ発生ステップと、

前記符号化ビデオデータ、前記現在の符号化処理で発生された現在の符号化パラメータ、並びに過去の符号化処理で生成された過去の符号化パラメータを含む MPEG ビットストリームを発生する MPEG ビットストリーム発生ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 102】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化装置において、

現在の符号化処理として、前記ビデオデータを符号化し、符号化ビデオデータを生成する符号化手段と、

シーケンスレイヤ、GOP レイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有する MPEG ビットストリームを生成するストリーム生成手段とを備え、

前記各レイヤは、前記現在の符号化処理で生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを備えること特徴とするビデオ符号化装置。

【請求項 103】 ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化方法において、

現在の符号化処理として、符号化ビデオデータを生成する符号化ビデオデータ生成ステップと、

シーケンスレイヤ、GOP レイヤ、ピクチャレイヤ、スライスレイヤ、並びにマクロブロックレイヤを含む MPEG ビットストリームを生成する MPEG ビットストリーム生成ステップとを含み、

前記各レイヤは、前記現在の符号化処理において生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを特徴とするビデオ符号化方法。

【請求項 104】 入力ビデオデータを符号化し、符号

化ビデオストリームを生成するビデオ符号化装置において、

過去の符号化処理で生成された過去の符号化パラメータを参照することにより、前記入力ビデオデータを符号化し、前記符号化ビデオストリームを生成する符号化手段と、

前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述手段と、

前記過去の符号化パラメータが記述されている前記符号化ビデオストリームを出力する出力手段とを備えること特徴とするビデオ符号化装置。

【請求項 105】 入力ビデオデータを符号化し、符号化ビデオストリームを生成するビデオ符号化方法において、

過去の符号化処理で生成された過去の符号化パラメータを参照することにより、前記入力ビデオデータを符号化し、前記符号化ビデオストリームを生成する生成ステップと、

前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述ステップと、

前記過去のパラメータが記述されている前記符号化ビデオストリームを出力する出力ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 106】 入力ビデオデータを符号化するビデオ符号化装置において、

前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報を受け取る受け取り手段と、

前記ヒストリ情報を参照して、前記入力ビデオデータを符号化する符号化手段と、

前記ヒストリ情報が将来の符号化処理において利用できるように、前記ヒストリ情報を前記符号化ストリームに記述する記述手段とを備えることを特徴とするビデオ符号化装置。

【請求項 107】 入力ビデオデータを符号化するビデオ符号化方法において、

前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報を受け取る受け取りステップと、

前記ヒストリ情報を参照することにより、前記入力ビデオデータを符号化する符号化ステップと、

前記ヒストリ情報が将来の符号化処理において利用できるように、前記符号化ストリームに前記ヒストリ情報を記述する記述ステップとを含むことを特徴とするビデオ符号化方法。

【請求項 108】 入力符号化ビデオストリームを処理するストリーム処理装置において、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

10

20

30

40

50

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

生成された前記過去の符号化パラメータを受け取り、前記過去の符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするストリーム処理装置。

【請求項 109】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号し、復号ビデオデータ

を生成する復号ステップと、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

生成された過去の符号化パラメータを受け取り、前記過去の符号化パラメータに基づいて、前記符号化ステップの処理における前記現在の符号化処理を制御する制御ステップとを含むこと特徴とするストリーム処理方法。

【請求項 110】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出し、前記過去の符号化パラメータをヒストリ情報として出力する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段

と、過去の符号化パラメータを含む前記ヒストリ情報を受け取り、前記現在の符号化処理が、前記過去の符号化パラメータを選択的に使用することで最適化されるように、前記ヒストリ情報に基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするストリーム処理システム。

【請求項 111】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、

ヒストリ情報として前記過去の符号化パラメータを出力する出力ステップと、現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

過去の符号化パラメータを含む前記ヒストリ情報を受け取る受け取りステップと、

前記現在の符号化処理が、前記過去の符号化パラメータを選択的に使用することで最適化されるように、前記ヒストリ情報に基づいて、前記符号化ステップの処理における現在の符号化処理を制御する制御ステップとを含むこと特徴とするストリーム処理方法。

【請求項 112】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

前記過去の符号化パラメータを受け取り、前記過去の符号化パラメータから前記現在の符号化処理に対応する最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするストリーム処理システム。

【請求項 113】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

前記過去の符号化パラメータを受け取り、前記過去の符号化パラメータから前記現在の符号化処理に対応する最適な符号化パラメータを選択する受け取りステップと、前記最適な符号化パラメータに基づいて、前記符号化ステップの前記現在の符号化処理を制御する制御ステップとを含むこと特徴とするストリーム処理方法。

【請求項 114】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する復号手段と、

割り当てられたピクチャタイプで前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、

過去の符号化処理において生成された過去の符号化パラメータを受け取り、前記割り当てられたピクチャタイプに従って、前記過去の符号化パラメータから最適な符号化パラメータを選択し、前記最適な符号化パラメータに基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするストリーム処理シ

ステム。

【請求項 115】 入力符号化ビデオストリームを処理するストリーム処理方法において、前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、前記入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する抽出ステップと、

割り当てられたピクチャタイプで前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、

過去の符号化処理で生成された過去の符号化パラメータを受け取り、前記割り当てられたピクチャタイプに従って、前記過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、

前記最適な符号化パラメータに基づいて、前記符号化ステップの現在の符号化処理を制御する制御ステップとを含むこと特徴とするストリーム処理方法。

【請求項 116】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、

割り当てられたピクチャタイプで前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、

過去の符号化処理において、割り当てられたピクチャタイプで前記参照ピクチャが符号化されたか否かを判定し、その判定結果に基づいて、前記符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とするストリーム処理システム。

【請求項 117】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、

割り当てられたピクチャタイプで前記復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、

過去の符号化処理において、前記割り当てられたピクチャタイプで前記参照ピクチャが符号化されたか否かを判定する判定ステップと、

前記判定ステップの処理での前記判定結果に基づいて、前記符号化ステップでの現在の符号化処理を制御する制御ステップとを含むこと特徴とするストリーム処理方法。

【請求項 118】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

前記符号化ビデオデータ、前記現在の符号化処理で生成された現在の符号化パラメータ、並びに過去の符号化処理で生成された過去の符号化パラメータを含むMPEGストリームを生成するストリーム生成手段とを備えること特徴とするストリーム処理システム。

【請求項 119】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

前記符号化ビデオデータ、前記現在の符号化処理で生成された現在の符号化パラメータ、並びに過去の符号化処理で生成された過去の符号化パラメータを含むMPEGストリームを生成するストリーム生成ステップとを含むこと特徴とするストリーム処理方法。

【請求項 120】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームを生成するストリーム生成手段とを備え、

前記各レイヤは、前記現在の符号化処理で生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを備えること特徴とするストリーム処理システム。

【請求項 121】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理

10

20

30

40

50

で生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、並びにマクロブロックレイヤを有するMPEGビットストリームを生成する生成ステップとを含み、前記各レイヤは、前記現在の符号化処理で生成された現在の符号化パラメータを含み、前記ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを特徴とするストリーム処理方法。

【請求項 1 2 2】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成し、前記入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、

現在の符号化処理として、前記過去の符号化パラメータを参照して、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、

前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述手段と、

前記過去の符号化パラメータが記述されている前記符号化ビデオストリームを出力する出力手段とを備えること特徴とするストリーム処理システム。

【請求項 1 2 3】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ステップと、

前記入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、

現在の符号化処理として、前記過去の符号化パラメータを参照して、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

前記過去の符号化パラメータを前記符号化ビデオストリームに記述する記述ステップと、

前記過去の符号化パラメータが記述されている前記符号化ビデオストリームを出力する出力ステップとを含むこと特徴とするストリーム処理方法。

【請求項 1 2 4】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号手段と、

前記入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パラメータを含む履歴情報を抽出する抽出手段と、

現在の符号化処理として、前記履歴情報を参照して、前記復号ビデオデータを符号化し、符号化ビデオ

トリームを生成する符号化手段と、

前記履歴情報が将来の符号化処理で利用できるように、前記履歴情報を前記符号化ストリームに記述する記述手段とを備えること特徴とするストリーム処理システム。

【請求項 1 2 5】 入力符号化ビデオストリームを処理するストリーム処理方法において、

前記入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、

10 前記入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パラメータを含む履歴情報を抽出する抽出ステップと、

現在の符号化処理として、前記履歴情報を参照して、前記復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、

前記履歴情報が将来の符号化処理で利用できるように、前記履歴情報を前記符号化ストリームに記述する記述ステップとを含むこと特徴とするストリーム処理方法。

20 【請求項 1 2 6】 入力符号化ビデオストリームを処理するストリーム処理システムにおいて、

前記入力符号化ビデオストリームを直近の符号化処理の符号化パラメータに基づいて復号してベースバンドビデオデータを生成し、過去の符号化処理における過去の符号化パラメータを抽出し、前記直近の符号化処理、および過去の符号化処理の前記符号化パラメータを前記ベースバンドビデオデータに多重化する復号手段と、

30 新たな符号化ビデオストリームのビットレートまたはGOP構造が、前記入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、前記ベースバンドビデオデータを前記直近の符号化処理と過去の符号化処理における前記符号化パラメータに基づいて符号化し、前記新たな符号化ビデオストリームを生成する符号化手段とを備えること特徴とするストリーム処理システム。

【請求項 1 2 7】 入力符号化ビデオストリームを復号するビデオ復号方法において、

前記入力符号化ビデオストリームのシンタックスを解析し、過去の符号化処理で生成された過去の符号化パラメータを抽出する抽出ステップと、

40 前記符号化ビデオストリームを復号する復号ステップと、

前記過去の符号化パラメータが、前記復号ビデオデータの将来の符号化処理において、利用できるように、前記復号ビデオデータと、前記過去の符号化パラメータを出力する出力ステップとを含むことを特徴とするビデオ復号方法。

【請求項 1 2 8】 入力符号化ビデオストリームを復号するビデオ復号装置において、

50 前記入力符号化ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータ

と、過去の符号化処理で生成された過去の符号化パラメータを抽出する解析手段と、

前記現在の符号化パラメータに基づいて、前記入力符号化ビデオストリームを復号してビデオデータを生成する復号手段と、

前記現在の符号化パラメータと過去の符号化パラメータの両方が前記復号ビデオデータの将来の符号化処理において、利用できるように、前記復号ビデオデータ、現在の符号化パラメータ、および過去の符号化パラメータを出力する出力手段とを備えることを特徴とするビデオ復号装置。

【請求項 129】 入力符号化ビデオストリームを復号するビデオ復号方法において、

前記入力復号ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータと、過去の符号化処理で生成された過去の符号化パラメータを抽出する解析ステップと、

前記現在の符号化パラメータに基づいて、前記符号化ビデオストリームを復号してビデオデータを生成する復号ステップと、

前記現在の符号化パラメータと過去の符号化パラメータの両方が、前記復号ビデオデータの将来の符号化処理において、利用できるように、前記復号ビデオデータ、現在の符号化パラメータ、並びに過去の符号化パラメータを出力する出力ステップとを含むことを特徴とするビデオ復号方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、符号化ストリーム記録装置および方法、符号化ストリーム再生装置および方法、トランスコーディングシステム、変換装置および方法、ビデオ符号化装置および方法、ストリーム処理装置および方法、ストリーム処理システム、ビデオ復号装置および方法、並びに記録媒体に関し、特に、MPEG規格に基づいて符号化された符号化ビットストリームのGOP

(Group of Pictures) の構造を変更したり、符号化ビットストリームのビットレートを変更するためのトランスコーディング装置に用いて好適な符号化ストリーム記録装置および方法、符号化ストリーム再生装置および方法、トランスコーディングシステム、変換装置および方法、ビデオ符号化装置および方法、ストリーム処理装置および方法、ストリーム処理システム、ビデオ復号装置および方法、並びに記録媒体に関する。

【0002】

【従来の技術】近年、テレビジョンプログラムを制作及び放送する放送局においては、ビデオデータを圧縮/符号化処理するために、MPEG (Moving Picture Experts Group) 技術が一般的に使われるようになってきた。特に、ビデオデータをテープなどのランダムアクセス可能な記録媒体素材に記録する場合、及びビデオデータをケ

ーブルや衛星を介して伝送する場合には、このMPEG技術がデファクトスタンダードになりつつある。

【0003】放送局において制作されたビデオプログラムが各家庭に伝送されるまでの放送局における処理の一例を簡単に説明する。まず、ビデオカメラとVTR (Video Tape Recorder) が一体となったカムコーダに設けられたエンコーダによって、ソースビデオデータをエンコード処理して磁気テープ上に記録する。この際、カムコーダのエンコーダは、VTRのテープの記録フォーマットに適するように、ソースビデオデータを符号化する。たとえば、この磁気テープ上に記録されるMPEGビットストリームのGOP構造は、2フレームから1GOPが構成される構造 (たとえば、I, B, I, B, I, B, ……) とされる。また磁気テープ上に記録されているMPEGビットストリームのビットレートは、18Mbpsである。

【0004】次に、メイン放送局において、この磁気テープ上に記録されたビデオビットストリームを編集する編集処理を行う。そのために、磁気テープ上に記録されたビデオビットストリームのGOP構造を、編集処理に適したGOP構造に変換する。編集処理に適したGOP構造とは、1GOPが1フレームから構成され、すべてのピクチャがIピクチャであるGOP構造である。なぜなら、フレーム単位で編集を行うためには、他のピクチャと相関のないIピクチャがもっとも適しているからである。実際のオペレーションとしては、磁気テープ上に記録されたビデオストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオ信号を、すべてのピクチャがIピクチャとなるように再エンコードする。このようにデコード処理及び再エンコード処理を行うことによって、編集処理に適したGOP構造を有したビットストリームを生成することができる。

【0005】次に、上述した編集処理によって生成された編集ビデオプログラムを、メイン局から地方局に伝送するために、編集ビデオプログラムのビットストリームを、伝送処理に適したGOP構造及びビットレートに変換する。放送局間の伝送に適したGOP構造とは、たとえば、1GOPが15フレームから構成されているGOP構造 (たとえば、I, B, B, P, B, B, P…) である。また、放送局間の伝送に適したビットレートは、一般的に放送局間においては、光ファイバなどの高伝送容量を有した専用線が設けられているので、50Mbps以上のハイビットレートであることが望ましい。具体的には、編集処理されたビデオプログラムのビットストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオデータを上述した放送局間の伝送に適したGOP構造及びビットレートを有するように再エンコードする。

【0006】地方局においては、メイン局から伝送されてきたビデオプログラムの中に、地方特有の商業的挿入のために編集処理が行われる。つまり、上述

した編集処理と同じように、メイン局から伝送されてきたビデオストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオ信号を、すべてのピクチャがIピクチャとなるように再エンコードすることによって、編集処理に適したGOP構造を有したビットストリームを生成することができる。

【0007】続いて、この地方局において編集処理が行われたビデオプログラムを各家庭に、ケーブルや衛星を介して伝送するために、この伝送処理に適したGOP構造及びビットレートに変換する。たとえば、各家庭に伝送するための伝送処理に適したGOP構造とは、1GOPが15フレームから構成されるGOP構造（たとえば、I、B、B、P、B、B、P……）であって、各家庭に伝送するための伝送処理に適したビットレートは、5Mbps程度の低ビットレートである。具体的には、編集処理されたビデオプログラムのビットストリームを一旦デコードしてベースバンドのビデオデータに戻す。そして、そのベースバンドのビデオデータを上述した伝送処理に適したGOP構造及びビットレートを有するように再エンコードする。

【0008】

【発明が解決しようとする課題】以上の説明からも理解できるように、放送局から各家庭にビデオプログラムが伝送される間に、複数回の復号処理及び符号化処理が繰り返されている。実際には、放送局における処理は上述した信号処理以外にもさまざまな信号処理が必要であり、そのたびに復号処理及び符号化処理を繰り返さなければならない。

【0009】しかしながら、MPEG規格に基づく符号化処理及び復号処理は、100%可逆の処理ではないことは良く知られている。つまり、エンコードされる前のベースバンドのビデオデータと、デコードされた後のビデオデータは100%同じでは無く、この符号化処理及び復号処理によって画質が劣化している。つまり、上述したように、デコード処理及びエンコード処理を繰り返すと、その処理の度に、画質が劣化してしまうと言う問題があった。別の言葉で表現すると、デコード/エンコード処理を繰り返す毎に、画質の劣化が蓄積されてしまう。

【0010】本発明は、このような状況に鑑みてなされたものであり、MPEG規格に基づいて符号化された符号化ビットストリームのGOP (Group of Pictures) の構造を変更するために復号及び符号化処理を繰り返したとしても画質劣化の発生しないトランスコーディングシステムを実現できるようにするものである。

【0011】

【課題を解決するための手段】請求項1に記載の符号化ストリーム記録装置は、入力された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する

検出手段と、検出手段の検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化手段と、符号化手段により生成された第2の符号化ストリームと、検出手段により検出された過去の符号化処理における符号化履歴とを、記録媒体に記録する記録手段とを備えることを特徴とする。

【0012】前記符号化手段には、MPEG方式で符号化を行わせることができる。

【0013】前記記録手段には、符号化履歴を、MPEG方式の第2の符号化ストリームのuser_dataとして多重化して記録させることができる。

【0014】前記記録手段には、符号化履歴を、第2の符号化ストリームに較べて、同期コードからより離れた位置に多重化して記録させることができる。

【0015】請求項5に記載の符号化ストリーム記録方法は、入力された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、検出ステップの処理における検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化ステップと、符号化ステップの処理により生成された第2の符号化ストリームと、検出ステップの処理により検出された過去の符号化処理における符号化履歴とを、記録媒体に記録する記録ステップとを含むことを特徴とする。

【0016】請求項6に記載の記録媒体のプログラムは、入力された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、検出ステップの処理における検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化ステップと、符号化ステップの処理により生成された第2の符号化ストリームと、検出ステップの処理により検出された過去の符号化処理における符号化履歴とを、記録媒体に記録する記録ステップとを含むことを特徴とする。

【0017】請求項1に記載の符号化ストリーム記録装置、請求項5に記載の符号化ストリーム記録方法、および請求項6に記載の記録媒体のプログラムにおいては、入力された第1の符号化ストリームの過去の符号化処理における符号化履歴が検出され、その検出結果を利用して、第2の符号化ストリームが生成され、生成された第2の符号化ストリームと、検出された過去の符号化処理における符号化履歴とが、記録媒体に記録される。

【0018】請求項7に記載の符号化ストリーム再生装置は、記録媒体に記録されている第1の符号化ストリームを再生する再生手段と、再生手段により再生された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出手段と、検出手段の検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化手段と、符号化手段により生成された第2の符号化ストリームと、検出手段に

より検出された過去の符号化処理における符号化履歴とを、多重化して出力する出力手段とを備えることを特徴とする。

【0019】前記符号化手段には、第1の符号化ストリームをMPEG方式で符号化させることができる。

【0020】前記検出手段には、符号化履歴を、MPEG方式の第1の符号化ストリームのuser_dataから検出させることができる。また、検出手段には、符号化履歴を、第1の符号化ストリームに較べて、同期コードからより離れた位置から検出させることができる。

【0021】請求項11に記載の符号化ストリーム再生方法は、記録媒体に記録されている第1の符号化ストリームを再生する再生ステップと、再生ステップの処理により再生された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、検出ステップの処理における検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化ステップと、符号化ステップの処理により生成された第2の符号化ストリームと、検出ステップの処理により検出された過去の符号化処理における符号化履歴とを、多重化して出力する出力ステップとを含むことを特徴とする。

【0022】請求項12に記載の記録媒体のプログラムは、記録媒体に記録されている第1の符号化ストリームを再生する再生ステップと、再生ステップの処理により再生された第1の符号化ストリームの過去の符号化処理における符号化履歴を検出する検出ステップと、検出ステップの処理における検出結果を利用して、第1の符号化ストリームを符号化して第2の符号化ストリームを生成する符号化ステップと、符号化ステップの処理により生成された第2の符号化ストリームと、前記検出ステップの処理により検出された過去の符号化処理における符号化履歴とを、多重化して出力する出力ステップとを含むことを特徴とする。

【0023】請求項7に記載の符号化ストリーム再生装置、請求項11に記載の符号化ストリーム再生方法、および請求項12に記載の記録媒体のプログラムにおいて、記録媒体から再生された第1の符号化ストリームの過去の符号化処理における符号化履歴が検出され、その検出結果を利用して、第2の符号化ストリームが生成され、生成された前記第2の符号化ストリームと、検出された過去の符号化処理における符号化履歴とが、多重化して出力される。

【0024】請求項13に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号してビデオデータを生成するとともに、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、過去の符号化パラメー

タを受け取り、過去の符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0025】請求項13に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号してビデオデータが生成されるとともに、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、現在の符号化処理が制御される。

【0026】前記符号化手段には、現在の符号化処理において、入力ビデオデータに含まれる参照ピクチャに割り当てられた現在のピクチャタイプで、参照ピクチャをエンコードさせ、制御手段には、過去の符号化処理において割り当てられたピクチャタイプと同一のピクチャタイプに参照ピクチャが符号化されたか否かを判定し、その判定結果に基づいて、現在の符号化処理を制御させるようにすることができる。

【0027】前記符号化手段には、ビデオデータに含まれる参照ピクチャを、現在の符号化処理において、参照ピクチャに割り当てられた現在のピクチャタイプで符号化させ、制御手段には、現在の符号化処理において、参照ピクチャに割り当てられた過去のピクチャタイプを、過去の符号化パラメータを参照することで検出し、現在のピクチャタイプと過去のピクチャタイプに基づいて現在の符号化処理を制御させるようにすることができる。

【0028】前記制御手段には、判定に基づいて、過去の符号化パラメータから最適な符号化パラメータを選択し、選択された最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0029】前記制御手段には、参照ピクチャを、過去の符号化処理において発生された過去の符号化パラメータの1つを使用して符号化させるようにすることができる。

【0030】前記過去の符号化パラメータには、過去の符号化処理において生成された動きベクトル情報を含ませ、符号化手段には、現在の符号化処理において、参照ピクチャの動きベクトル情報を検出するための動きベクトル検出手段を含ませるようにすることができる。

【0031】前記制御手段には、判断の結果に基づいて、動きベクトル検出手段の動作を制御させるようにすることができる。

【0032】前記制御手段には、動きベクトル検出手段における新たな動きベクトル情報の計算の代わりに、過去の符号化パラメータに含まれる、動きベクトル情報を再使用させるようにすることができる。

【0033】前記参照ピクチャが過去の符号化処理にお

10

20

30

40

50

いて、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、過去の符号化処理に含まれる動きベクトル情報を再使用させるようにすることができる。

【0034】前記参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプで符号化されていない場合、制御手段には、新たな動きベクトル情報が、動きベクトル検出手段により検出されたように動きベクトル検出手段を制御させるようにすることができる。

【0035】前記制御手段には、過去の符号化パラメータから、現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0036】前記符号化手段には、現在の符号化処理において、参照ピクチャに割り当てられた現在のピクチャタイプで入力ビデオデータに含まれる参照ピクチャを符号化させ、制御手段には、参照ピクチャが過去の符号化処理において割り当てられたピクチャタイプと同一のピクチャタイプに符号化されたか否かを判定し、その判定結果に基づいて、最適な符号化パラメータを選択させるようにすることができる。

【0037】前記過去の符号化パラメータには、フレーム予測モードまたはフィールド予測モードを表す予測モード情報を含ませ、制御手段には、予測モード情報に対応して、現在の符号化処理を制御させるようにすることができる。

【0038】前記参照ピクチャが、過去の符号化処理において、過去のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、新たな予測モード情報の計算に代えて、過去の符号化パラメータに含まれる予測モード情報を再使用させるようにすることができる。

【0039】前記符号化パラメータには、イントラ予測、前方予測、後方予測、または双方向予測を示す予測タイプ情報を含ませ、制御手段には、予測タイプ情報に基づいて、現在の符号化処理を制御させるようにすることができる。

【0040】前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、新たな予測タイプ情報の計算に代えて、過去の符号化パラメータに含まれる予測タイプ情報を再使用させるようにすることができる。

【0041】前記符号化パラメータには、フレームDCTモードまたはフィールドDCTモードを表すDCTモード情報を含ませ、制御手段には、DCTモード情報に基づいて、現在の符号化処理を制御させるようにすることができる。

【0042】前記参照ピクチャが過去の符号化処理にお

いて、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、新たなDCTモード情報の計算に代えて、過去の符号化パラメータに含まれる、DCTモード情報を再使用させるようにすることができる。

【0043】前記制御手段には、符号化手段の現在の符号化処理に対応する現在の符号化パラメータを生成させるようにすることができる。

【0044】前記制御手段には、現在の符号化パラメータと過去の符号化パラメータから、現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0045】前記符号化手段には、入力ビデオデータに含まれる参照ピクチャを、現在の符号化処理において、参照ピクチャに割り当てられた現在のピクチャタイプで符号化させ、制御手段には、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて最適な符号化パラメータを選択させるようにすることができる。

【0046】前記過去の符号化パラメータには、過去の符号化処理において生成された量子化情報を含ませ、符号化手段には、現在の符号化処理において、参照ピクチャを量子化する量子化手段を含ませるようにすることができる。

【0047】前記制御手段には、符号化ビデオストリームを蓄積する伝送バッファの残量を表すバッファ情報を受け取り、バッファ情報に基づいて、伝送バッファのオーバーフローおよびアンダーフローを阻止するように、量子化手段を制御させるようにすることができる。

【0048】前記制御手段には、バッファ情報から抽出した量子化ステップサイズ、および過去の符号化パラメータに含まれる量子化情報から抽出した量子化ステップサイズに基づいて、量子化手段を制御させるようにすることができる。

【0049】前記制御手段には、バッファ情報に対応する量子化ステップサイズと、量子化情報に対応する量子化ステップサイズから選択した最も大きい量子化ステップサイズを使用して、量子化手段を制御させるようにすることができる。

【0050】前記制御手段には、符号化手段が過去の符号化パラメータを符号化ビデオストリームに記述するように制御させるようにすることができる。

【0051】前記符号化手段には、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有するMPEG標準に従ったMPEGビットストリームを、符号化ビデオストリームを処理して生成する処理手段を持たせるようにすることができる。

【0052】前記制御手段には、符号化手段の現在の符

号化処理に対応する現在の符号化パラメータを生成し、処理手段には、現在の符号化パラメータをピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤに記述し、過去の符号化パラメータをピクチャレイヤのユーザデータエリアに記述させるようにすることができる。

【0053】前記処理手段には、ユーザデータエリアに、過去の符号化パラメータを記述するために、過去の符号化パラメータを含むヒストリストリームを生成させるようにすることができる。

【0054】前記処理手段には、MPEG標準に規定されているスタートコードのエミレーションを防止するために、ヒストリストリームにマーカビットを挿入し、マーカビットが挿入されたヒストリストリームをピクチャレイヤのユーザデータエリアに記述させるようにすることができる。

【0055】請求項43に記載の変換方法は、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する生成ステップと、入力符号化ビデオストリームから過去の符号化処理において生成された符号化パラメータを抽出する抽出ステップと、現在の符号化処理により、符号化ビデオストリームを生成するように、ビデオデータを符号化する符号化ステップと、生成された過去の符号化パラメータを受け取る受け取りステップと、過去の符号化パラメータに基づいて、符号化ステップの処理の現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0056】請求項43に記載の変換方法においては、入力符号化ビデオストリームを復号し、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理において生成された符号化パラメータが抽出され、現在の符号化処理により、符号化ビデオストリームを生成するように、ビデオデータが符号化され、生成された過去の符号化パラメータが受け取られ、過去の符号化パラメータに基づいて、現在の符号化処理が制御される。

【0057】請求項44に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出し、過去の符号化パラメータをヒストリ情報として出力する復号手段と、復号されたビデオデータを符号化し、符号化ビデオストリームを現在の符号化処理として生成する符号化手段と、過去の符号化パラメータを含む、ヒストリ情報を受け取り、過去の符号化パラメータを選択的に使用することで、現在の符号化処理が最適化されるように、ヒストリ情報に基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0058】請求項44に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復

号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、過去の符号化パラメータをヒストリ情報として出力され、復号されたビデオデータが符号化され、符号化ビデオストリームが現在の符号化処理として生成され、過去の符号化パラメータを含む、ヒストリ情報が受け取られ、過去の符号化パラメータを選択的に使用することで、現在の符号化処理が最適化されるように、ヒストリ情報に基づいて、現在の符号化処理が制御される。

【0059】請求項45に記載の変換方法は、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、過去の符号化パラメータをヒストリ情報として出力する出力ステップと、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する生成ステップと、過去の符号化パラメータを含むヒストリ情報を受け取る受け取りステップと、現在の符号化処理が、過去の符号化パラメータを選択的に使用することで最適化されるように、ヒストリ情報に基づいて、符号化ステップ処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0060】請求項45に記載の変換方法においては、入力符号化ビデオストリームを復号し、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、過去の符号化パラメータがヒストリ情報として出力され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを含むヒストリ情報が受け取られ、現在の符号化処理が、過去の符号化パラメータを選択的に使用することで最適化されるように、ヒストリ情報に基づいて、現在の符号化処理が制御される。

【0061】請求項46に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0062】請求項46に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオ

オストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0063】請求項47に記載の変換方法は、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する生成ステップと、過去の符号化パラメータを受け取る受け取りステップと、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択する選択ステップと、最適な符号化パラメータに基づいて、符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0064】請求項47に記載の変換方法においては、入力符号化ビデオストリームを復号し、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを受け取られ、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータが選択され、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0065】請求項48に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、割り当てられたピクチャタイプで、符号化ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、過去の符号化処理において生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0066】請求項48に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで、符号化ビデオデータに含まれる参照ピ

クチャが符号化され、過去の符号化処理において生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0067】請求項49に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する生成ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、割り当てられたピクチャタイプで、符号化ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、過去の符号化処理において生成された過去の符号化パラメータを受け取る受け取りステップと、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、最適な符号化パラメータに基づいて、符号化ステップの処理での現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0068】請求項49に記載の変換方法においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで、符号化ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理において生成された過去の符号化パラメータを受け取られ、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータが選択され、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0069】請求項50に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、割り当てられたピクチャタイプで、復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0070】請求項50に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで、復号ビデオデータに含まれる参照ピクチャが符号化され、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて、現在の符号

化処理が制御される。

【0071】請求項51に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、割り当てられたピクチャタイプで、復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かを判定する判定ステップと、判定の結果に基づいて、符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0072】請求項51に記載の変換方法においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで、復号ビデオデータに含まれる参照ピクチャが符号化され、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かが判定され、判定の結果に基づいて、現在の符号化処理が制御される。

【0073】請求項52に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、符号化ビデオストリーム、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームを生成するストリーム生成手段とを備えることを特徴とする。

【0074】請求項52に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータが符号化され、符号化ビデオストリームが生成され、符号化ビデオストリーム、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームが生成される。

【0075】請求項53に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、入力符号化ビデオストリームから、過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符号化処理として、復号ビデオデータを符号

化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、符号化ビデオストリーム、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームを生成するMPEGストリーム生成ステップとを含むことを特徴とする。

【0076】請求項53に記載の変換方法においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理において生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、符号化ビデオストリーム、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGストリームが生成される。

【0077】請求項54に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームを生成するストリーム生成手段とを備え、各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理において生成された過去の符号化パラメータを含むことを特徴とする。

【0078】請求項54に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームが生成される。各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理において生成された過去の符号化パラメータを含む。

【0079】請求項55に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符号化処理として、復号ビデオデータを符号化

し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームを生成するMPEGビットストリーム生成ステップとを含み、各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理において生成された過去の符号化パラメータを含むことを特徴とする。

【0080】請求項55に記載の変換方法においては、10
入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを含むMPEGビットストリームが生成される。各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理におい20
て生成された過去の符号化パラメータを含む。

【0081】請求項56に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、過去の符号化パラメータを参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、過去の符号化パラメータを符号化ビデオストリームに記述する記述手段と、過去の符号化パラメータが記述されている符号化ビデオストリームを出力する出力手段とを備えることを特徴とする。30

【0082】請求項56に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、過去の符号化パラメータを参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータが符号化ビデオストリームに記述され、過去の符号化パラメータが記述されている符号化ビデオストリームが出力される。40

【0083】請求項57に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、過去の符号化パラメータを参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、過去の50

符号化パラメータを符号化ビデオストリームに記述する記述ステップと、過去のパラメータが記述されている符号化ビデオストリームを出力する出力ステップとを含むことを特徴とする。

【0084】請求項57に記載の変換方法においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、過去の符号化パラメータを参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータが符号化ビデオストリームに記述され、過去のパラメータが記述されている符号化ビデオストリームが出力される。

【0085】請求項58に記載のトランスコーディングシステムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成手段と、入力符号化ビデオストリームから、過去の符号化処理により生成された以前の符号化パラメータを含むヒストリ情報を抽出する抽出手段と、ヒストリ情報を参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成手段と、ヒストリ情報が、将来の符号化処理において利用できるように、符号化ストリームに記述する記述手段とを備えることを特徴とする。

【0086】請求項58に記載のトランスコーディングシステムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された以前の符号化パラメータを含むヒストリ情報が抽出され、ヒストリ情報を参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、ヒストリ情報が、将来の符号化処理において利用できるように、符号化ストリームに記述される。

【0087】請求項59に記載の変換方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ビデオデータ生成ステップと、入力符号化ビデオストリームから、過去の符号化処理により生成された前の符号化パラメータを含むヒストリ情報を抽出する抽出ステップと、ヒストリ情報を参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップと、ヒストリ情報が、将来の符号化処理において利用できるように符号化ストリームにヒストリ情報を記述する記述ステップとを含むことを特徴とする。

【0088】請求項59に記載の変換方法においては、入力符号化ビデオストリームを復号して、復号ビデオデ

ータが生成され、入力符号化ビデオストリームから、過去の符号化処理により生成された前の符号化パラメータを含む履歴情報が抽出され、履歴情報を参照することにより、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、履歴情報が、将来の符号化処理において利用できるように符号化ストリームに履歴情報が記述される。

【0089】請求項60に記載のトランスコーディングシステムは、直近の符号化処理の符号化パラメータに基づいて、入力符号化ビデオストリームを復号してベースバンドビデオデータを生成し、過去の符号化処理の過去の符号化パラメータを抽出し、直近および過去の符号化処理の符号化パラメータをベースバンドビデオデータに多重化する復号手段と、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、直近および過去の符号化処理の符号化パラメータに基づいて、ベースバンドビデオデータを符号化し、新たな符号化ビデオストリームを生成する符号化手段とを備えることを特徴とする。

【0090】請求項60に記載のトランスコーディングシステムにおいては、直近の符号化処理の符号化パラメータに基づいて、入力符号化ビデオストリームを復号してベースバンドビデオデータを生成し、過去の符号化処理の過去の符号化パラメータを抽出し、直近および過去の符号化処理の符号化パラメータがベースバンドビデオデータに多重化され、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、直近および過去の符号化処理の符号化パラメータに基づいて、ベースバンドビデオデータを符号化し、新たな符号化ビデオストリームが生成される。

【0091】請求項61に記載の変換方法は、直近の符号化処理の符号化パラメータに基づいて、入力符号化ビデオストリームを符号化して、ベースバンドビデオデータを生成するベースバンドビデオデータ生成ステップと、過去の符号化処理の過去の符号化パラメータを抽出する抽出ステップと、直近および過去の符号化処理の符号化パラメータをベースバンドビデオデータに多重化する多重化ステップと、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、直近および過去の符号化処理の符号化パラメータに基づいて、ベースバンドビデオデータを符号化して、新たな符号化ビデオストリームを生成する符号化ビデオストリーム生成ステップとを含むことを特徴とする。

【0092】請求項61に記載の変換方法においては、直近の符号化処理の符号化パラメータに基づいて、入力符号化ビデオストリームを符号化して、ベースバンドビデオデータが生成され、過去の符号化処理の過去の符号

化パラメータが抽出され、直近および過去の符号化処理の符号化パラメータがベースバンドビデオデータに多重化され、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、直近および過去の符号化処理の符号化パラメータに基づいて、ベースバンドビデオデータを符号化して、新たな符号化ビデオストリームが生成される。

【0093】請求項62に記載のビデオ符号化装置は、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化手段と、過去の符号化処理により生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0094】請求項62に記載のビデオ符号化装置においては、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームが生成され、過去の符号化処理により生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、現在の符号化処理が制御される。

【0095】前記符号化手段には、現在の符号化処理において、入力ビデオデータに含まれる参照ピクチャを、参照ピクチャに割り当てられた現在のピクチャタイプで符号化させ、制御手段には、参照ピクチャが過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、現在の符号化処理を判定結果に基づいて制御させるようにすることができる。

【0096】前記符号化手段には、現在の符号化処理において、入力ビデオデータに含まれる参照ピクチャを、割り当てられた現在のピクチャタイプで符号化させ、制御手段には、現在の符号化処理で、参照ピクチャに割り当てられた過去のピクチャタイプを、過去の符号化パラメータを参照することにより検出し、現在のピクチャタイプおよび過去のピクチャタイプに基づいて、現在の符号化処理を制御させるようにすることができる。

【0097】前記制御手段には、判定結果に従って、過去の符号化パラメータから最適な符号化パラメータを選択し、選択した最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0098】前記制御手段には、1つの過去の符号化処理で生成された過去の符号化パラメータを使用して参照ピクチャを符号化させるようにすることができる。

【0099】前記過去の符号化パラメータには、過去の符号化処理において生成された動きベクトルを含ませ、符号化手段には、現在の符号化処理において、参照ピクチャの動きベクトル情報を検出する動きベクトル検出手段を含ませるようにすることができる。

【0100】前記制御手段には、判定結果に基づいて動きベクトル検出手段の処理を制御させるようにすることができる。

【0101】前記制御手段には、動きベクトル検出手段において、新たな動きベクトル情報を計算する代わりに、過去の符号化パラメータに含まれる動きベクトル情報を再使用させるようにすることができる。

【0102】前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、過去の符号化処理に含まれる動きベクトル情報を再使用させるようにすることができる。

【0103】前記制御手段には、参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプで符号化されていない場合、新たな動きベクトル情報が、動きベクトル検出手段で検出されたように、動きベクトル検出手段を制御させるようにすることができる。

【0104】前記制御手段には、過去の符号化パラメータから、現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0105】前記符号化手段には、現在の符号化処理において、入力ビデオデータに含まれる参照ピクチャを、割り当てられた現在のピクチャタイプで符号化させ、制御手段には、参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、判定結果に基づいて、最適な符号化パラメータを選択させるようにすることができる。

【0106】前記過去の符号化パラメータには、フレーム予測モードまたはフィールド予測モードを表す予測モード情報を含ませ、制御手段には、予測モード情報に従って、現在の符号化処理を制御させるようにすることができる。

【0107】前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、新たな予測モード情報を計算する代わりに、過去の符号化パラメータに含まれる予測モード情報を再使用させるようにすることができる。

【0108】前記符号化パラメータには、イントラ予測、前方予測、後方予測、または両方向予測を表す予測タイプ情報を含ませ、制御手段には、予測タイプ情報に基づいて、現在の符号化処理を制御させるようにすることができる。

【0109】前記制御手段には、参照ピクチャが、現在のピクチャタイプと同一のピクチャタイプで過去の符号化処理において符号化されている場合、新たな予測タイプ情報を計算する代わりに、過去の符号化パラメータに含まれる予測タイプ情報を再使用させるようにすること

ができる。

【0110】前記符号化パラメータには、フレームDCTモードまたはフィールドDCTモードを表すDCTモード情報を含ませ、制御手段には、DCTモード情報に基づいて、現在の符号化処理を制御させるようにすることができる。

【0111】前記参照ピクチャが、過去の符号化処理において、現在のピクチャタイプと同一のピクチャタイプで符号化されている場合、制御手段には、新たなDCTモード情報を計算する代わりに、過去の符号化パラメータに含まれるDCTモード情報を再使用させるようにすることができる。

【0112】前記制御手段には、符号化手段の現在の符号化処理に対応する現在の符号化パラメータを生成させるようにすることができる。

【0113】前記制御手段には、現在の符号化パラメータと過去の符号化パラメータから、現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御させるようにすることができる。

【0114】前記符号化手段には、現在の符号化処理において、入力ビデオデータに含まれる参照ピクチャを、参照ピクチャに割り当てられた現在のピクチャタイプで符号化させ、制御手段には、参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプと同一のピクチャタイプで符号化されたか否かを判定し、その判定結果に基づいて、最適な符号化パラメータを選択させるようにすることができる。

【0115】前記過去の符号化パラメータには、過去の符号化処理で生成された量子化情報を含ませ、符号化手段には、現在の符号化処理において参照ピクチャを量子化する量子化手段を含ませるようにすることができる。

【0116】前記制御手段には、符号化ビデオストリームを蓄積する伝送バッファの残量を表すバッファ情報を受け取り、伝送バッファのオーバーフローおよびアンダーフローの発生を抑制するように、バッファ情報に基づいて量子化手段を制御させるようにすることができる。

【0117】前記制御手段には、バッファ情報から生成された量子化ステップサイズ、および過去の符号化パラメータに含まれる量子化情報から生成された量子化ステップサイズに基づいて、量子化手段を制御させるようにすることができる。

【0118】前記制御手段には、バッファ情報に対応する量子化ステップサイズ、および量子化情報に対応する量子化ステップサイズの中から選択された最大の量子化ステップサイズを利用して、量子化手段を制御させるようにすることができる。

【0119】前記制御手段には、符号化手段が過去の符号化パラメータを符号化ビデオストリームに記述するように符号化手段を制御させるようにすることができる。

【0120】前記符号化手段には、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有するMPEG標準に基づくMPEGビットストリームを生成するように、符号化ビデオストリームを処理する処理手段を含ませるようにすることができる。

【0121】前記制御手段には、符号化手段の現在の符号化処理に対応する現在の符号化パラメータを発生させ、符号化手段には、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤに現在の符号化パラメータを記述し、過去の符号化パラメータをピクチャレイヤのユーザデータエリアに記述させるようにすることができる。

【0122】前記符号化手段には、ユーザデータエリアに、過去の符号化パラメータを記述するために、過去の符号化パラメータを含むヒストリストリームを発生させるようにすることができる。

【0123】前記符号化手段には、MPEG標準に指定されているスタートコードのエミレーションの発生を防止するために、ヒストリストリームにマーカビットを挿入し、ピクチャレイヤのユーザデータエリアにマーカビットが挿入されたヒストリストリームを記述させるようにすることができる。

【0124】請求項92に記載のビデオ符号化装置は、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化手段と、過去の符号化処理で生成された過去の符号化パラメータを含むヒストリ情報を受け取り、現在の符号化処理が過去の符号化パラメータを選択的に使用することにより最適化されるように、ヒストリ情報に基づいて符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0125】請求項92に記載のビデオ符号化装置においては、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームが生成され、過去の符号化処理で生成された過去の符号化パラメータを含むヒストリ情報を受け取り、現在の符号化処理が過去の符号化パラメータを選択的に使用することにより最適化されるように、ヒストリ情報に基づいて現在の符号化処理が制御される。

【0126】請求項93に記載のビデオ符号化方法は、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームを生成する符号化ステップと、過去の符号化処理で発生された過去の符号化パラメータを含むヒストリ情報を受け取り、現在の符号化処理が過去の符号化パラメータを選択的に使用することで最適化されるように、ヒストリ情報に基づいて符号化ステップでの処理の現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0127】請求項93に記載のビデオ符号化方法にお

いては、現在の符号化処理として、入力ビデオデータを符号化して、符号化ビデオストリームが生成され、過去の符号化処理で発生された過去の符号化パラメータを含むヒストリ情報を受け取り、現在の符号化処理が過去の符号化パラメータを選択的に使用することで最適化されるように、ヒストリ情報に基づいて現在の符号化処理が制御される。

【0128】請求項94に記載のビデオ符号化装置は、現在の符号化処理として、ビデオデータを符号化して、符号化ビデオストリームを生成する符号化手段と、過去の符号化処理で発生された過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0129】請求項94に記載のビデオ符号化装置においては、現在の符号化処理として、ビデオデータを符号化して、符号化ビデオストリームが生成され、過去の符号化処理で発生された過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0130】請求項95に記載のビデオ符号化方法は、現在の符号化処理として、ビデオデータを符号化して、符号化ビデオストリームを生成する符号化ステップ、過去の符号化処理で発生された過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0131】請求項95に記載のビデオ符号化方法においては、現在の符号化処理として、ビデオデータを符号化して、符号化ビデオストリームが生成され、過去の符号化処理で発生された過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0132】請求項96に記載のビデオ符号化装置は、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに対応して、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0133】請求項96に記載のビデオ符号化装置にお

いては、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに対応して、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0134】請求項97に記載のビデオ符号化方法は、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、最適な符号化パラメータに基づいて、符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0135】請求項97に記載のビデオ符号化方法においては、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータが選択され、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0136】請求項98に記載のビデオ符号化装置は、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたかどうかを判定し、判定結果に基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0137】請求項98に記載のビデオ符号化装置においては、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたかどうかを判定し、判定結果に基づいて、現在の符号化処理が制御される。

【0138】請求項99に記載のビデオ符号化方法は、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かを判定する判定ステップと、判定ステップでの判定結果に基づいて、符号化ステップでの現在の符号化処理を制御する制御ステップとを含むことを特徴とする。

【0139】請求項99に記載のビデオ符号化方法においては、割り当てられたピクチャタイプで、ビデオデータに含まれる参照ピクチャが符号化され、参照ピクチャが、過去の符号化処理において、割り当てられたピクチャタイプで符号化されたか否かが判定され、判定結果に

基づいて、現在の符号化処理が制御される。

【0140】請求項100に記載のビデオ符号化装置は、現在の符号化処理として、入力ビデオデータを符号化し、符号化ビデオデータを生成する符号化手段と、符号化ビデオデータ、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGビットストリームを生成するストリーム生成手段とを備えることを特徴とする。

【0141】請求項100に記載のビデオ符号化装置においては、現在の符号化処理として、入力ビデオデータを符号化し、符号化ビデオデータが生成され、符号化ビデオデータ、現在の符号化処理において生成された現在の符号化パラメータ、並びに過去の符号化処理において生成された過去の符号化パラメータを含むMPEGビットストリームが生成される。

【0142】請求項101に記載のビデオ符号化方法は、現在の符号化処理として、符号化ビデオデータを発生する符号化ビデオデータ発生ステップと、符号化ビデオデータ、現在の符号化処理で発生された現在の符号化パラメータ、並びに過去の符号化処理で生成された過去の符号化パラメータを含むMPEGビットストリームを発生するMPEGビットストリーム発生ステップとを含むことを特徴とする。

【0143】請求項101に記載のビデオ符号化方法においては、現在の符号化処理として、符号化ビデオデータが発生され、符号化ビデオデータ、現在の符号化処理で発生された現在の符号化パラメータ、並びに過去の符号化処理で生成された過去の符号化パラメータを含むMPEGビットストリームが発生される。

【0144】請求項102に記載のビデオ符号化装置は、現在の符号化処理として、ビデオデータを符号化し、符号化ビデオデータを生成する符号化手段と、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有するMPEGビットストリームを生成するストリーム生成手段とを備え、各レイヤは、現在の符号化処理で生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを備えることを特徴とする。

【0145】請求項102に記載のビデオ符号化装置においては、現在の符号化処理として、ビデオデータを符号化し、符号化ビデオデータが生成され、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロックレイヤを有するMPEGビットストリームが生成される。各レイヤは、現在の符号化処理で生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含む。

【0146】請求項103に記載のビデオ符号化方法

は、現在の符号化処理として、符号化ビデオデータを生成する符号化ビデオデータ生成ステップと、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、並びにマクロブロックレイヤを含むMPEGビットストリームを生成するMPEGビットストリーム生成ステップとを含み、各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含むことを特徴とする。

【0147】請求項103に記載のビデオ符号化方法においては、現在の符号化処理として、符号化ビデオデータが生成され、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、並びにマクロブロックレイヤを含むMPEGビットストリームが生成され、各レイヤは、現在の符号化処理において生成された現在の符号化パラメータを含み、ピクチャレイヤはさらに、過去の符号化処理で生成された過去の符号化パラメータを含む。

【0148】請求項104に記載のビデオ符号化装置は、過去の符号化処理で生成された過去の符号化パラメータを参照することにより、入力ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、前記過去の符号化パラメータを符号化ビデオストリームに記述する記述手段と、過去の符号化パラメータが記述されている符号化ビデオストリームを出力する出力手段とを備えることを特徴とする。

【0149】請求項104に記載のビデオ符号化装置においては、過去の符号化処理で生成された過去の符号化パラメータを参照することにより、入力ビデオデータを符号化し、符号化ビデオストリームが生成され、前記過去の符号化パラメータが符号化ビデオストリームに記述され、過去の符号化パラメータが記述されている符号化ビデオストリームが出力される。

【0150】請求項105に記載のビデオ符号化方法は、過去の符号化処理で生成された過去の符号化パラメータを参照することにより、入力ビデオデータを符号化し、符号化ビデオストリームを生成する生成ステップと、過去の符号化パラメータを符号化ビデオストリームに記述する記述ステップと、過去のパラメータが記述されている符号化ビデオストリームを出力する出力ステップとを含むことを特徴とする。

【0151】請求項105に記載のビデオ符号化方法においては、過去の符号化処理で生成された過去の符号化パラメータを参照することにより、入力ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータが符号化ビデオストリームに記述され、過去のパラメータが記述されている符号化ビデオストリームが出力される。

【0152】請求項106に記載のビデオ符号化装置は、前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報を受け取る受け取り手段

と、ヒストリ情報を参照して、入力ビデオデータを符号化する符号化手段と、ヒストリ情報が将来の符号化処理において利用できるように、ヒストリ情報を符号化ストリームに記述する記述手段とを備えることを特徴とする。

【0153】請求項106に記載のビデオ符号化装置においては、前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報が受け取られ、ヒストリ情報を参照して、入力ビデオデータが符号化され、ヒストリ情報が将来の符号化処理において利用できるように、ヒストリ情報が符号化ストリームに記述される。

【0154】請求項107に記載のビデオ符号化方法は、前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報を受け取る受け取りステップと、ヒストリ情報を参照することにより、入力ビデオデータを符号化する符号化ステップと、前記ヒストリ情報が将来の符号化処理において利用できるように、符号化ストリームにヒストリ情報を記述する記述ステップとを含むことを特徴とする。

【0155】請求項107に記載のビデオ符号化方法においては、前の符号化処理において生成された複数の符号化パラメータを含むヒストリ情報が受け取られ、ヒストリ情報を参照することにより、入力ビデオデータが符号化され、前記ヒストリ情報が将来の符号化処理において利用できるように、符号化ストリームにヒストリ情報が記述される。

【0156】請求項108に記載のストリーム処理装置は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えることを特徴とする。

【0157】請求項108に記載のストリーム処理装置においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、現在の符号化処理が制御される。

【0158】請求項109に記載のストリーム処理方法は、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符

号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、符号化ステップの処理における現在の符号化処理を制御する制御ステップとを含むこと特徴とする。

【0159】請求項109に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号し、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、生成された過去の符号化パラメータを受け取り、過去の符号化パラメータに基づいて、現在の符号化処理が制御される。

【0160】請求項110に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出し、過去の符号化パラメータを履歴情報として出力する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、過去の符号化パラメータを含む履歴情報を受け取り、現在の符号化処理が、過去の符号化パラメータを選択的に使用することで最適化されるように、履歴情報に基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とする。

【0161】請求項110に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出し、過去の符号化パラメータが履歴情報として出力され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを含む履歴情報を受け取り、現在の符号化処理が、過去の符号化パラメータを選択的に使用することで最適化されるように、履歴情報に基づいて、現在の符号化処理が制御される。

【0162】請求項111に記載のストリーム処理方法は、入力符号化ビデオストリームを復号し、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する抽出ステップと、履歴情報として過去の符号化パラメータを出力する出力ステップと、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、過去の符号化パラメータを含む履歴情報を受け取る受け取りステップと、現在の符号化処理が、

過去の符号化パラメータを選択的に使用することで最適化されるように、履歴情報に基づいて、符号化ステップの処理における現在の符号化処理を制御する制御ステップとを含むこと特徴とする。

【0163】請求項111に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号し、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、履歴情報として過去の符号化パラメータが出力され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを含む履歴情報が受け取られ、現在の符号化処理が、過去の符号化パラメータを選択的に使用することで最適化されるように、履歴情報に基づいて、現在の符号化処理が制御される。

【0164】請求項112に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とする。

【0165】請求項112に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0166】請求項113に記載のストリーム処理方法は、入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータを選択する受け取りステップと、最適な符号化パラメータに基づいて、符号化ステップの現在の符号化処理を制御する制御ステップとを含むこと特徴とする。

【0167】請求項113に記載のストリーム処理方法

においては、入力符号化ビデオストリームを復号して復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータが抽出され、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータを受け取り、過去の符号化パラメータから現在の符号化処理に対応する最適な符号化パラメータが選択され、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0168】請求項114に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して、復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する復号手段と、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、過去の符号化処理において生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とする。

【0169】請求項114に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理において生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択し、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0170】請求項115に記載のストリーム処理方法は、入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する抽出ステップと、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータを選択する選択ステップと、最適な符号化パラメータに基づいて、符号化ステップの現在の符号化処理を制御する制御ステップとを含むこと特徴とする。

【0171】請求項115に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号して復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで復

号ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理で生成された過去の符号化パラメータを受け取り、割り当てられたピクチャタイプに従って、過去の符号化パラメータから最適な符号化パラメータが選択され、最適な符号化パラメータに基づいて、現在の符号化処理が制御される。

【0172】請求項116に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャを符号化する符号化手段と、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたか否かを判定し、その判定結果に基づいて、符号化手段の現在の符号化処理を制御する制御手段とを備えること特徴とする。

【0173】請求項116に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたか否かを判定し、その判定結果に基づいて、現在の符号化処理が制御される。

【0174】請求項117に記載のストリーム処理方法は、入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャを符号化する符号化ステップと、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたか否かを判定する判定ステップと、判定ステップの処理での判定結果に基づいて、符号化ステップでの現在の符号化処理を制御する制御ステップとを含むこと特徴とする。

【0175】請求項117に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号して復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータが抽出され、割り当てられたピクチャタイプで復号ビデオデータに含まれる参照ピクチャが符号化され、過去の符号化処理において、割り当てられたピクチャタイプで参照ピクチャが符号化されたか否かが判定され、判定結果に基づいて、現在の符号化処理が制御される。

【0176】請求項118に記載のストリーム処理シ

【0180】請求項120に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理で生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤ、スライスレイヤ、およびマクロブロッ

【0184】請求項122に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータを抽出する復号手段と、現在の符号化処理として、過去の符号化パラメータを参照して、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化

手段と、過去の符号化パラメータを符号化ビデオストリームに記述する記述手段と、過去の符号化パラメータが記述されている符号化ビデオストリームを出力する出力手段とを備えること特徴とする。

【0185】請求項122に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して復号ビデオデータを生成し、入力符号化ビデオストリームから過去の符号化処理により生成された過去の符号化パラメータが抽出され、現在の符号化処理として、過去の符号化パラメータを参照して、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータが符号化ビデオストリームに記述され、過去の符号化パラメータが記述されている符号化ビデオストリームが出力される。

【0186】請求項123に記載のストリーム処理方法は、入力符号化ビデオストリームを復号して、復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータを抽出する抽出ステップと、現在の符号化処理として、過去の符号化パラメータを参照して、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、過去の符号化パラメータを符号化ビデオストリームに記述する記述ステップと、過去の符号化パラメータが記述されている符号化ビデオストリームを出力する出力ステップとを含むこと特徴とする。

【0187】請求項123に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号して、復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理において生成された過去の符号化パラメータが抽出され、現在の符号化処理として、過去の符号化パラメータを参照して、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、過去の符号化パラメータが符号化ビデオストリームに記述され、過去の符号化パラメータが記述されている符号化ビデオストリームが出力される。

【0188】請求項124に記載のストリーム処理システムは、入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号手段と、入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パラメータを含むヒストリ情報を抽出する抽出手段と、現在の符号化処理として、ヒストリ情報を参照して、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化手段と、ヒストリ情報が将来の符号化処理で利用できるように、ヒストリ情報を符号化ストリームに記述する記述手段とを備えること特徴とする。

【0189】請求項124に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを復号して復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パ

ラメータを含むヒストリ情報が抽出され、現在の符号化処理として、ヒストリ情報を参照して、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、ヒストリ情報が将来の符号化処理で利用できるように、ヒストリ情報が符号化ストリームに記述される。

【0190】請求項125に記載のストリーム処理方法は、入力符号化ビデオストリームを復号して復号ビデオデータを生成する復号ステップと、入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パラメータを含むヒストリ情報を抽出する抽出ステップと、現在の符号化処理として、ヒストリ情報を参照して、復号ビデオデータを符号化し、符号化ビデオストリームを生成する符号化ステップと、ヒストリ情報が将来の符号化処理で利用できるように、ヒストリ情報を符号化ストリームに記述する記述ステップとを含むこと特徴とする。

【0191】請求項125に記載のストリーム処理方法においては、入力符号化ビデオストリームを復号して復号ビデオデータが生成され、入力符号化ビデオストリームから過去の符号化処理で生成された前の符号化パラメータを含むヒストリ情報が抽出され、現在の符号化処理として、ヒストリ情報を参照して、復号ビデオデータを符号化し、符号化ビデオストリームが生成され、ヒストリ情報が将来の符号化処理で利用できるように、ヒストリ情報が符号化ストリームに記述される。

【0192】請求項126に記載のストリーム処理システムは、入力符号化ビデオストリームを直近の符号化処理の符号化パラメータに基づいて復号してベースバンドビデオデータを生成し、過去の符号化処理における過去の符号化パラメータを抽出し、直近の符号化処理、および過去の符号化処理の符号化パラメータをベースバンドビデオデータに多重化する復号手段と、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、ベースバンドビデオデータを直近の符号化処理と過去の符号化処理における符号化パラメータに基づいて符号化し、新たな符号化ビデオストリームを生成する符号化手段とを備えること特徴とする。

【0193】請求項126に記載のストリーム処理システムにおいては、入力符号化ビデオストリームを直近の符号化処理の符号化パラメータに基づいて復号してベースバンドビデオデータを生成し、過去の符号化処理における過去の符号化パラメータを抽出し、直近の符号化処理、および過去の符号化処理の符号化パラメータがベースバンドビデオデータに多重化され、新たな符号化ビデオストリームのビットレートまたはGOP構造が、入力符号化ビデオストリームのビットレートまたはGOP構造と異なるように、ベースバンドビデオデータを直近の符号化処理と過去の符号化処理における符号化パラメータに基づいて符号化し、新たな符号化ビデオストリームが生

成される。

【0194】請求項127に記載のビデオ復号方法は、入力符号化ビデオストリームのシンタックスを解析し、過去の符号化処理で生成された過去の符号化パラメータを抽出する抽出ステップと、符号化ビデオストリームを復号する復号ステップと、過去の符号化パラメータが、復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータと、過去の符号化パラメータを出力する出力ステップとを含むことを特徴とする。

【0195】請求項127に記載のビデオ復号方法においては、入力符号化ビデオストリームのシンタックスを解析し、過去の符号化処理で生成された過去の符号化パラメータが抽出され、符号化ビデオストリームが復号され、過去の符号化パラメータが、復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータと、過去の符号化パラメータが出力される。

【0196】請求項128に記載のビデオ復号装置は、入力符号化ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータと、過去の符号化処理で生成された過去の符号化パラメータを抽出する解析手段と、現在の符号化パラメータに基づいて、入力符号化ビデオストリームを復号してビデオデータを生成する復号手段と、現在の符号化パラメータと過去の符号化パラメータの両方が復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータ、現在の符号化パラメータ、および過去の符号化パラメータを出力する出力手段とを備えることを特徴とする。

【0197】請求項128に記載のビデオ復号装置においては、入力符号化ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータと、過去の符号化処理で生成された過去の符号化パラメータが抽出され、現在の符号化パラメータに基づいて、入力符号化ビデオストリームを復号してビデオデータが生成され、現在の符号化パラメータと過去の符号化パラメータの両方が復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータ、現在の符号化パラメータ、および過去の符号化パラメータが出力される。

【0198】請求項129に記載のビデオ復号方法は、入力復号ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータと、過去の符号化処理で生成された過去の符号化パラメータを抽出する解析ステップと、現在の符号化パラメータに基づいて、符号化ビデオストリームを復号してビデオデータを生成する復号ステップと、現在の符号化パラメータと過去の符号化パラメータの両方が、復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータ、現在の符号化パラメータ、並び

に過去の符号化パラメータを出力する出力ステップとを含むことを特徴とする。

【0199】請求項129に記載のビデオ復号方法においては、入力復号ビデオストリームのシンタックスを解析し、直近の符号化処理で生成された現在の符号化パラメータと、過去の符号化処理で生成された過去の符号化パラメータが抽出され、現在の符号化パラメータに基づいて、符号化ビデオストリームを復号してビデオデータが生成され、現在の符号化パラメータと過去の符号化パラメータの両方が、復号ビデオデータの将来の符号化処理において、利用できるように、復号ビデオデータ、現在の符号化パラメータ、並びに過去の符号化パラメータが出力される。

【0200】

【発明の実施の形態】以下に、本発明を適用したトランスコーダについて説明するが、その前に、動画像信号の圧縮符号化について説明する。なお、本明細書においてシステムの用語は、複数の装置、手段などにより構成される全体的な装置を意味するものである。

【0201】例えば、テレビ会議システム、テレビ電話システムなどのように、動画像信号を遠隔地に伝送するシステムにおいては、伝送路を効率良く利用するため、映像信号のライン相関やフレーム間相関を利用して、画像信号を圧縮符号化するようになされている。

【0202】ライン相関を利用すると、画像信号を、例えばDCT（離散コサイン変換）処理するなどして圧縮することができる。

【0203】また、フレーム間相関を利用すると、画像信号をさらに圧縮して符号化することが可能となる。例えば図1に示すように、時刻 t_1 乃至 t_3 において、フレーム画像PC1乃至PC3がそれぞれ発生している場合、フレーム画像PC1およびPC2の画像信号の差を演算して、PC12を生成し、また、フレーム画像PC2およびPC3の差を演算して、PC23を生成する。通常、時間的に隣接するフレームの画像は、それ程大きな変化を有していないため、両者の差を演算すると、その差分信号は小さな値のものとなる。そこで、この差分信号を符号化すれば、符号量を圧縮することができる。

【0204】しかしながら、差分信号のみを伝送したのでは、元の画像を復元することができない。そこで、各フレームの画像を、Iピクチャ、PピクチャまたはBピクチャの3種類のピクチャタイプのいずれかとし、画像信号を圧縮符号化するようにしている。

【0205】すなわち、例えば図2に示すように、フレームF1乃至F17までの17フレームの画像信号をグループオブピクチャ(GOP)とし、処理の1単位とする。そして、その先頭のフレームF1の画像信号はIピクチャとして符号化し、第2番目のフレームF2はBピクチャとして、また第3番目のフレームF3はPピクチャとして、それぞれ処理する。以下、第4番目以降のフレ

ムF4乃至F17は、BピクチャまたはPピクチャとして交互に処理する。

【0206】Iピクチャの画像信号としては、その1フレーム分の画像信号をそのまま伝送する。これに対して、Pピクチャの画像信号としては、基本的には、図2に示すように、それより時間的に先行するIピクチャまたはPピクチャの画像信号からの差分を伝送する。さらにBピクチャの画像信号としては、基本的には、図3に示すように、時間的に先行するフレームまたは後行するフレームの両方の平均値からの差分を求め、その差分を符号化する。

【0207】図4は、このようにして、動画像信号を符号化する方法の原理を示している。同図に示すように、最初のフレームF1は、Iピクチャとして処理されるため、そのまま伝送データF1Xとして伝送路に伝送される(画像内符号化)。これに対して、第2のフレームF2は、Bピクチャとして処理されるため、時間的に先行するフレームF1と、時間的に後行するフレームF3の平均値との差分が演算され、その差分が伝送データF2Xとして伝送される。

【0208】ただし、このBピクチャとしての処理は、さらに細かく説明すると、4種類存在する。その第1の処理は、元のフレームF2のデータをそのまま伝送データF2Xとして伝送するものであり(SP1)(イントラ符号化)、Iピクチャにおける場合と同様の処理となる。第2の処理は、時間的に後のフレームF3からの差分を演算し、その差分(SP2)を伝送するものである(後方予測符号化)。第3の処理は、時間的に先行するフレームF1との差分(SP3)を伝送するものである(前方予測符号化)。さらに第4の処理は、時間的に先行するフレームF1と後行するフレームF3の平均値との差分(SP4)を生成し、これを伝送データF2Xとして伝送するものである(両方向予測符号化)。

【0209】実際には、上述した4つの方法のうちの伝送データが最も少なくなる方法が採用される。

【0210】なお、差分データを伝送するとき、差分を演算する対象となるフレームの画像(参照画像)との間の動きベクトル x_1 (フレームF1とF2の間の動きベクトル)(前方予測の場合)、もしくは x_2 (フレームF3とF2の間の動きベクトル)(後方予測の場合)、または x_1 と x_2 の両方(両方向予測の場合)が、差分データとともに伝送される。

【0211】また、PピクチャのフレームF3は、時間的に先行するフレームF1を参照画像として、このフレームとの差分信号(SP3)と、動きベクトル x_3 が演算され、これが伝送データF3Xとして伝送される(前方予測符号化)。あるいはまた、元のフレームF3のデータが、そのままデータF3Xとして伝送される(SP1)(イントラ符号化)。これらの方法のうち、Bピクチャにおける場合と同様に、伝送データがより少なくな

る方法が選択される。

【0212】図5は、上述した原理に基づいて、動画像信号を符号化して伝送し、これを復号する装置の構成例を示している。符号化装置1は、入力された映像信号を符号化し、伝送路としての記録媒体3に伝送するようになされている。そして、復号装置2は、記録媒体3に記録された信号を再生し、これを復号して出力するようになされている。

【0213】符号化装置1においては、入力された映像信号が前処理回路11に入力され、そこで輝度信号と色信号(本実施の形態の場合、色差信号)が分離され、それぞれA/D変換器12、13でアナログ信号からデジタル信号に変換される。A/D変換器12、13によりデジタル信号に変換された映像信号は、フレームメモリ14に供給され、記憶される。フレームメモリ14は、輝度信号を輝度信号フレームメモリ15に、また、色差信号を色差信号フレームメモリ16に、それぞれ記憶させる。

【0214】フォーマット変換回路17は、フレームメモリ14に記憶されたフレームフォーマットの信号を、ブロックフォーマットの信号に変換する。すなわち、図6に示すように、フレームメモリ14に記憶された映像信号は、1ライン当りHドットのラインがVライン集められた、図6(A)に示すようなフレームフォーマットのデータとされている。フォーマット変換回路17は、この1フレームの信号を、図6(B)に示すように、16ラインを単位としてM個のスライスに区分する。そして、各スライスは、M個のマクロブロックに分割される。マクロブロックは、図6(C)に示すように、 16×16 個の画素(ドット)に対応する輝度信号により構成され、この輝度信号は、さらに 8×8 ドットを単位とするブロックY[1]乃至Y[4]に区分される。そして、この 16×16 ドットの輝度信号には、 8×8 ドットのCb信号と、 8×8 ドットのCr信号が対応される。

【0215】このように、ブロックフォーマットに変換されたデータは、フォーマット変換回路17からエンコーダ18に供給され、ここでエンコード(符号化)が行われる。その詳細については、図7を参照して後述する。

【0216】エンコーダ18によりエンコードされた信号は、ビットストリームとして伝送路に出力される。例えば記録回路19に供給され、デジタル信号として記録媒体3に記録される。

【0217】復号装置2の再生回路30により記録媒体3より再生されたデータは、デコーダ31に供給され、デコードされる。デコーダ31の詳細については、図12を参照して後述する。

【0218】デコーダ31によりデコードされたデータは、フォーマット変換回路32に入力され、ブロックフォーマットからフレームフォーマットに変換される。そ

して、フレームフォーマットの輝度信号は、フレームメモリ 33 の輝度信号フレームメモリ 34 に供給されて記憶され、色差信号は色差信号フレームメモリ 35 に供給されて記憶される。輝度信号フレームメモリ 34 と色差信号フレームメモリ 35 から読み出された輝度信号と色差信号は、それぞれ D/A 変換器 36、37 によりアナログ信号に変換され、後処理回路 38 に供給される。後処理回路 38 は、輝度信号と色差信号を合成して出力する。

【0219】次に図 7 を参照して、エンコーダ 18 の構成について説明する。符号化される画像データは、マクロブロック単位で動きベクトル検出回路 50 に入力される。動きベクトル検出回路 50 は、予め設定されている所定のシーケンスに従って、各フレームの画像データを、I ピクチャ、P ピクチャ、または B ピクチャとして処理する。シーケンシャルに入力される各フレームの画像を、I、P、または B のいずれのピクチャとして処理するかは、予め定められている（例えば、図 2 と図 3 に示したように、フレーム F1 乃至 F17 により構成されるグループオブピクチャが、I、B、P、B、P、 \dots B、P として処理される）。

【0220】I ピクチャとして処理されるフレーム（例えば、フレーム F1）の画像データは、動きベクトル検出回路 50 からフレームメモリ 51 の前方原画像部 51a に転送、記憶され、B ピクチャとして処理されるフレーム（例えば、フレーム F2）の画像データは、原画像部 51b に転送、記憶され、P ピクチャとして処理されるフレーム（例えば、フレーム F3）の画像データは、後方原画像部 51c に転送、記憶される。

【0221】また、次のタイミングにおいて、さらに B ピクチャ（フレーム F4）または P ピクチャ（フレーム F5）として処理すべきフレームの画像が入力されたとき、それまで後方原画像部 51c に記憶されていた最初の P ピクチャ（フレーム F3）の画像データが、前方原画像部 51a に転送され、次の B ピクチャ（フレーム F4）の画像データが、参照原画像部 51b に記憶（上書き）され、次の P ピクチャ（フレーム F5）の画像データが、後方原画像部 51c に記憶（上書き）される。このような動作が順次繰り返される。

【0222】フレームメモリ 51 に記憶された各ピクチャの信号は、そこから読み出され、予測モード切り替え回路 52 において、フレーム予測モード処理、またはフィールド予測モード処理が行なわれる。

【0223】さらにまた、予測判定回路 54 の制御の下に、演算器 53 において、画像内予測、前方予測、後方予測、または両方向予測の演算が行なわれる。これらの処理のうち、いずれの処理を行なうかは、予測誤差信号（処理の対象とされている参照画像と、これに対する予測画像との差分）に対応して決定される。このため、動きベクトル検出回路 50 は、この判定に用いられる予測

誤差信号の絶対値和（自乗和でもよい）を生成する。

【0224】ここで、予測モード切り替え回路 52 におけるフレーム予測モードとフィールド予測モードについて説明する。

【0225】フレーム予測モードが設定された場合においては、予測モード切り替え回路 52 は、動きベクトル検出回路 50 より供給される 4 個の輝度ブロック Y

【1】乃至 Y【4】を、そのまま後段の演算器 53 に出力する。すなわち、この場合においては、図 8 に示すように、各輝度ブロックに奇数フィールドのラインのデータと、偶数フィールドのラインのデータとが混在した状態となっている。このフレーム予測モードにおいては、4 個の輝度ブロック（マクロブロック）を単位として予測が行われ、4 個の輝度ブロックに対して 1 個の動きベクトルが対応される。

【0226】これに対して、予測モード切り替え回路 52 は、フィールド予測モードにおいては、図 8 に示す構成で動きベクトル検出回路 50 より入力される信号を、図 9 に示すように、4 個の輝度ブロックのうち、輝度ブロック Y【1】と Y【2】を、例えば奇数フィールドのラインのドットだけで構成させ、他の 2 個の輝度ブロック Y【3】と Y【4】を、偶数フィールドのラインのドットだけで構成させて、演算器 53 に出力する。この場合においては、2 個の輝度ブロック Y【1】と Y【2】に対して、1 個の動きベクトルが対応され、他の 2 個の輝度ブロック Y【3】と Y【4】に対して、他の 1 個の動きベクトルが対応される。

【0227】動きベクトル検出回路 50 は、フレーム予測モードにおける予測誤差の絶対値和、およびフィールド予測モードにおける予測誤差の絶対値和を予測モード切り替え回路 52 に出力する。予測モード切り替え回路 52 は、フレーム予測モードとフィールド予測モードにおける予測誤差の絶対値和を比較し、その値が小さい予測モードに対応する処理を施して、データを演算器 53 に出力する。

【0228】ただし、このような処理は、実際には動きベクトル検出回路 50 で行われる。すなわち、動きベクトル検出回路 50 は、決定されたモードに対応する構成の信号を予測モード切り替え回路 52 に出力し、予測モード切り替え回路 52 は、その信号を、そのまま後段の演算器 53 に出力する。

【0229】なお、色差信号は、フレーム予測モードの場合、図 8 に示すように、奇数フィールドのラインのデータと偶数フィールドのラインのデータとが混在する状態で、演算器 53 に供給される。また、フィールド予測モードの場合、図 9 に示すように、各色差ブロック Cb、Cr の上半分（4 ライン）が、輝度ブロック Y

【1】、Y【2】に対応する奇数フィールドの色差信号とされ、下半分（4 ライン）が、輝度ブロック Y【3】、Y【4】に対応する偶数フィールドの色差信号

とされる。

【0230】また、動きベクトル検出回路50は、以下に示すようにして、予測判定回路54において、画像内予測、前方予測、後方予測、または両方向予測のいずれの予測を行なうかを決定するための予測誤差の絶対値和を生成する。

【0231】すなわち、画像内予測の予測誤差の絶対値和として、参照画像のマクロブロックの信号 A_{ij} の総和 ΣA_{ij} の絶対値 $|\Sigma A_{ij}|$ と、マクロブロックの信号 A_{ij} の絶対値 $|A_{ij}|$ の総和 $\Sigma |A_{ij}|$ の差を求める。また、前方予測の予測誤差の絶対値和として、参照画像のマクロブロックの信号 A_{ij} と、予測画像のマクロブロックの信号 B_{ij} の差 $A_{ij}-B_{ij}$ の絶対値 $|A_{ij}-B_{ij}|$ の総和 $\Sigma |A_{ij}-B_{ij}|$ を求める。また、後方予測と両方向予測の予測誤差の絶対値和も、前方予測における場合と同様に（その予測画像を前方予測における場合と異なる予測画像に変更して）求める。

【0232】これらの絶対値和は、予測判定回路54に供給される。予測判定回路54は、前方予測、後方予測および両方向予測の予測誤差の絶対値和のうちの最も小さいものを、インタ予測の予測誤差の絶対値和として選択する。さらに、このインタ予測の予測誤差の絶対値和と、画像内予測の予測誤差の絶対値和とを比較し、その小さい方を選択し、この選択した絶対値和に対応するモードを予測モードとして選択する。すなわち、画像内予測の予測誤差の絶対値和の方が小さければ、画像内予測モードが設定される。インタ予測の予測誤差の絶対値和の方が小さければ、前方予測、後方予測または両方向予測モードのうちの対応する絶対値和が最も小さかったモードが設定される。

【0233】このように、動きベクトル検出回路50は、参照画像のマクロブロックの信号を、フレームまたはフィールド予測モードのうち、予測モード切り替え回路52により選択されたモードに対応する構成で、予測モード切り替え回路52を介して演算器53に供給するとともに、4つの予測モードのうちの予測判定回路54により選択された予測モードに対応する予測画像と参照画像の間の動きベクトルを検出し、可変長符号化回路58と動き補償回路64に出力する。上述したように、この動きベクトルとしては、対応する予測誤差の絶対値和が最小となるものが選択される。

【0234】予測判定回路54は、動きベクトル検出回路50が前方原画像部51aよりIピクチャの画像データを読み出しているとき、予測モードとして、フレームまたはフィールド（画像）内予測モード（動き補償を行わないモード）を設定し、演算器53のスイッチ53dを接点a側に切り替える。これにより、Iピクチャの画像データがDCTモード切り替え回路55に入力される。

【0235】DCTモード切り替え回路55は、図10または図11に示すように、4個の輝度ブロックのデータ

を、奇数フィールドのラインと偶数フィールドのラインが混在する状態（フレームDCTモード）、または、分離された状態（フィールドDCTモード）、のいずれかの状態にして、DCT回路56に出力する。

【0236】すなわち、DCTモード切り替え回路55は、奇数フィールドと偶数フィールドのデータを混在してDCT処理した場合における符号化効率と、分離した状態においてDCT処理した場合の符号化効率とを比較し、符号化効率の良好なモードを選択する。

【0237】例えば、入力された信号を、図10に示すように、奇数フィールドと偶数フィールドのラインが混在する構成とし、上下に隣接する奇数フィールドのラインの信号と偶数フィールドのラインの信号の差を演算し、さらにその絶対値の和（または自乗和）を求める。

【0238】また、入力された信号を、図11に示すように、奇数フィールドと偶数フィールドのラインが分離した構成とし、上下に隣接する奇数フィールドのライン同士の信号の差と、偶数フィールドのライン同士の信号の差を演算し、それぞれの絶対値の和（または自乗和）を求める。

【0239】さらに、両者（絶対値和）を比較し、小さい値に対応するDCTモードを設定する。すなわち、前者の方が小さければ、フレームDCTモードを設定し、後者の方が小さければ、フィールドDCTモードを設定する。

【0240】そして、選択したDCTモードに対応する構成のデータをDCT回路56に出力するとともに、選択したDCTモードを示すDCTフラグを、可変長符号化回路58、および動き補償回路64に出力する。

【0241】予測モード切り替え回路52における予測モード（図8と図9）と、このDCTモード切り替え回路55におけるDCTモード（図10と図11）を比較して明らかなように、輝度ブロックに関しては、両者の各モードにおけるデータ構造は実質的に同一である。

【0242】予測モード切り替え回路52において、フレーム予測モード（奇数ラインと偶数ラインが混在するモード）が選択された場合、DCTモード切り替え回路55においても、フレームDCTモード（奇数ラインと偶数ラインが混在するモード）が選択される可能性が高く、また予測モード切り替え回路52において、フィールド予測モード（奇数フィールドと偶数フィールドのデータが分離されたモード）が選択された場合、DCTモード切り替え回路55において、フィールドDCTモード（奇数フィールドと偶数フィールドのデータが分離されたモード）が選択される可能性が高い。

【0243】しかしながら、必ずしも常にこのようにモードが選択されるわけではなく、予測モード切り替え回路52においては、予測誤差の絶対値和が小さくなるようにモードが決定され、DCTモード切り替え回路55においては、符号化効率が良好となるようにモードが決定される。

【0244】DCTモード切り替え回路55より出力されたIピクチャの画像データは、DCT回路56に入力されてDCT処理され、DCT係数に変換される。このDCT係数は、量子化回路57に入力され、送信バッファ59のデータ蓄積量（バッファ蓄積量）に対応した量子化スケールで量子化された後、可変長符号化回路58に入力される。

【0245】可変長符号化回路58は、量子化回路57より供給される量子化スケール（スケール）に対応して、量子化回路57より供給される画像データ（いまの場合、Iピクチャのデータ）を、例えばハフマン符号などの可変長符号に変換し、送信バッファ59に出力する。

【0246】可変長符号化回路58にはまた、量子化回路57より量子化スケール（スケール）、予測判定回路54より予測モード（画像内予測、前方予測、後方予測、または両方向予測のいずれが設定されたかを示すモード）、動きベクトル検出回路50より動きベクトル、予測モード切り替え回路52より予測フラグ（フレーム予測モードまたはフィールド予測モードのいずれが設定されたかを示すフラグ）、およびDCTモード切り替え回路55が出力するDCTフラグ（フレームDCTモードまたはフィールドDCTモードのいずれが設定されたかを示すフラグ）が入力されており、これらも可変長符号化される。

【0247】送信バッファ59は、入力されたデータを一時蓄積し、蓄積量に対応するデータを量子化回路57に出力する。送信バッファ59は、そのデータ残量が許容上限値まで増量すると、量子化制御信号によって量子化回路57の量子化スケールを大きくすることにより、量子化データのデータ量を低下させる。また、これとは逆に、データ残量が許容下限値まで減少すると、送信バッファ59は、量子化制御信号によって量子化回路57の量子化スケールを小さくすることにより、量子化データのデータ量を増大させる。このようにして、送信バッファ59のオーバーフローまたはアンダフローが防止される。

【0248】そして、送信バッファ59に蓄積されたデータは、所定のタイミングで読み出され、伝送路に出力され、例えば記録回路19を介して記録媒体3に記録される。

【0249】一方、量子化回路57より出力されたIピクチャのデータは、逆量子化回路60に入力され、量子化回路57より供給される量子化スケールに対応して逆量子化される。逆量子化回路60の出力は、IDCT（逆離散コサイン変換）回路61に入力され、逆離散コサイン変換処理された後、演算器62を介してフレームメモリ63の前方予測画像部63a供給されて記憶される。

【0250】動きベクトル検出回路50は、シーケンスに入力される各フレームの画像データを、たとえ

ば、I、B、P、B、P、B・・・のピクチャとしてそれぞれ処理する場合、最初に入力されたフレームの画像データをIピクチャとして処理した後、次に入力されたフレームの画像をBピクチャとして処理する前に、さらにその次に入力されたフレームの画像データをPピクチャとして処理する。Bピクチャは、後方予測を伴うため、後方予測画像としてのPピクチャが先に用意されていないと、復号することができないからである。

【0251】そこで動きベクトル検出回路50は、Iピクチャの処理の次に、後方原画像部51cに記憶されているPピクチャの画像データの処理を開始する。そして、上述した場合と同様に、マクロブロック単位でのフレーム間差分（予測誤差）の絶対値和が、動きベクトル検出回路50から予測モード切り替え回路52と予測判定回路54に供給される。予測モード切り替え回路52と予測判定回路54は、このPピクチャのマクロブロックの予測誤差の絶対値和に対応して、フレーム/フィールド予測モード、または画像内予測、前方予測、後方予測、もしくは両方向予測の予測モードを設定する。

【0252】演算器53は、画像内予測モードが設定されたとき、スイッチ53dを上述したように接点a側に切り替える。したがって、このデータは、Iピクチャのデータと同様に、DCTモード切り替え回路55、DCT回路56、量子化回路57、可変長符号化回路58、および送信バッファ59を介して伝送路に伝送される。また、このデータは、逆量子化回路60、IDCT回路61、および演算器62を介してフレームメモリ63の後方予測画像部63bに供給されて記憶される。

【0253】また、前方予測モードが設定された場合、スイッチ53dが接点bに切り替えられるとともに、フレームメモリ63の前方予測画像部63aに記憶されている画像（いまの場合、Iピクチャの画像）データが読み出され、動き補償回路64により、動きベクトル検出回路50が出力する動きベクトルに対応して動き補償される。すなわち、動き補償回路64は、予測判定回路54より前方予測モードの設定が指令されたとき、前方予測画像部63aの読み出しアドレスを、動きベクトル検出回路50が、現在、出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0254】動き補償回路64より出力された予測画像データは、演算器53aに供給される。演算器53aは、予測モード切り替え回路52より供給された参照画像のマクロブロックのデータから、動き補償回路65より供給された、このマクロブロックに対応する予測画像データを減算し、その差分（予測誤差）を出力する。この差分データは、DCTモード切り替え回路55、DCT回路56、量子化回路57、可変長符号化回路58、および送信バッファ59を介して伝送路に伝送される。また、この差分データは、逆量子化回路60、およびIDCT回路

61により局所的に復号され、演算器62に入力される。

【0255】この演算器62にはまた、演算器53aに供給されている予測画像データと同一のデータが供給されている。演算器62は、IDCT回路61が出力する差分データに、動き補償回路64が出力する予測画像データを加算する。これにより、元の（復号した）Pピクチャの画像データが得られる。このPピクチャの画像データは、フレームメモリ63の後方予測画像部63bに供給されて記憶される。

【0256】動きベクトル検出回路50は、このように、IピクチャとPピクチャのデータが前方予測画像部63aと後方予測画像部63bにそれぞれ記憶された後、次にBピクチャの処理を実行する。予測モード切り替え回路52と予測判定回路54は、マクロブロック単位でのフレーム間差分の絶対値和の大きさに対応して、フレーム／フィールドモードを設定し、また、予測モードを画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのいずれかに設定する。

【0257】上述したように、画像内予測モードまたは前方予測モードの時、スイッチ53dは接点aまたはbに切り替えられる。このとき、Pピクチャにおける場合と同様の処理が行われ、データが伝送される。

【0258】これに対して、後方予測モードまたは両方向予測モードが設定された時、スイッチ53dは、接点cまたはdにそれぞれ切り替えられる。

【0259】スイッチ53dが接点cに切り替えられている後方予測モードの時、後方予測画像部63bに記憶されている画像（いまの場合、Pピクチャの画像）データが読み出され、動き補償回路64により、動きベクトル検出回路50が出力する動きベクトルに対応して動き補償される。すなわち、動き補償回路64は、予測判定回路54より後方予測モードの設定が指令されたとき、後方予測画像部63bの読み出しアドレスを、動きベクトル検出回路50が、現在、出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0260】動き補償回路64より出力された予測画像データは、演算器53bに供給される。演算器53bは、予測モード切り替え回路52より供給された参照画像のマクロブロックのデータから、動き補償回路64より供給された予測画像データを減算し、その差分を出力する。この差分データは、DCTモード切り替え回路55、DCT回路56、量子化回路57、可変長符号化回路58、および送信バッファ59を介して伝送路に伝送される。

【0261】スイッチ53dが接点dに切り替えられている両方向予測モードの時、前方予測画像部63aに記憶されている画像（いまの場合、Iピクチャの画像）デ

ータと、後方予測画像部63bに記憶されている画像（いまの場合、Pピクチャの画像）データが読み出され、動き補償回路64により、動きベクトル検出回路50が出力する動きベクトルに対応して動き補償される。

【0262】すなわち、動き補償回路64は、予測判定回路54より両方向予測モードの設定が指令されたとき、前方予測画像部63aと後方予測画像部63bの読み出しアドレスを、動きベクトル検出回路50がいま出力しているマクロブロックの位置に対応する位置から動きベクトル（この場合の動きベクトルは、前方予測画像用と後方予測画像用の2つとなる）に対応する分だけずらしてデータを読み出し、予測画像データを生成する。

【0263】動き補償回路64より出力された予測画像データは、演算器53cに供給される。演算器53cは、動きベクトル検出回路50より供給された参照画像のマクロブロックのデータから、動き補償回路64より供給された予測画像データの平均値を減算し、その差分を出力する。この差分データは、DCTモード切り替え回路55、DCT回路56、量子化回路57、可変長符号化回路58、および送信バッファ59を介して伝送路に伝送される。

【0264】Bピクチャの画像は、他の画像の予測画像とされることがないため、フレームメモリ63には記憶されない。

【0265】なお、フレームメモリ63において、前方予測画像部63aと後方予測画像部63bは、必要に応じてバンク切り替えが行われ、所定の参照画像に対して、一方または他方に記憶されているものを、前方予測画像あるいは後方予測画像として切り替えて出力することができる。

【0266】上述した説明においては、輝度ブロックを中心として説明をしたが、色差ブロックについても同様に、図8乃至図11に示すマクロブロックを単位として処理されて伝送される。なお、色差ブロックを処理する場合の動きベクトルは、対応する輝度ブロックの動きベクトルを垂直方向と水平方向に、それぞれ1/2にしたものが用いられる。

【0267】図12は、図5のデコーダ31の構成を示すブロック図である。伝送路（記録媒体3）を介して伝送された符号化された画像データは、図示せぬ受信回路で受信されたり、再生装置で再生され、受信バッファ81に一時記憶された後、復号回路90の可変長復号回路82に供給される。可変長復号回路82は、受信バッファ81より供給されたデータを可変長復号し、動きベクトル、予測モード、予測フラグ、およびDCTフラグを動き補償回路87に出力し、量子化スケールを逆量子化回路83に出力するとともに、復号された画像データを逆量子化回路83に出力する。

【0268】逆量子化回路83は、可変長復号回路82より供給された画像データを、同じく可変長復号回路8

2より供給された量子化スケールに従って逆量子化し、IDCT回路84に出力する。逆量子化回路83より出力されたデータ(DCT係数)は、IDCT回路84により、逆離散コサイン変換処理が施され、演算器85に供給される。

【0269】IDCT回路84より演算器85に供給された画像データが、Iピクチャのデータである場合、そのデータは演算器85より出力され、演算器85に後に入力される画像データ(PまたはBピクチャのデータ)の予測画像データ生成のために、フレームメモリ86の前方予測画像部86aに供給されて記憶される。また、このデータは、フォーマット変換回路32(図5)に出力される。

【0270】IDCT回路84より供給された画像データが、その1フレーム前の画像データを予測画像データとするPピクチャのデータであり、前方予測モードのデータである場合、フレームメモリ86の前方予測画像部86aに記憶されている、1フレーム前の画像データ(Iピクチャのデータ)が読み出され、動き補償回路87で可変長復号回路82より出力された動きベクトルに対応する動き補償が施される。そして、演算器85において、IDCT回路84より供給された画像データ(差分のデータ)と加算され、出力される。この加算されたデータ、すなわち、復号されたPピクチャのデータは、演算器85に後に入力される画像データ(BピクチャまたはPピクチャのデータ)の予測画像データ生成のために、フレームメモリ86の後方予測画像部86bに供給されて記憶される。

【0271】Pピクチャのデータであっても、画像内予測モードのデータは、Iピクチャのデータと同様に、演算器85において処理は行われず、そのまま後方予測画像部86bに記憶される。

【0272】このPピクチャは、次のBピクチャの次に表示されるべき画像であるため、この時点では、まだフォーマット変換回路32へ出力されない(上述したように、Bピクチャの後に入力されたPピクチャが、Bピクチャより先に処理され、伝送されている)。

【0273】IDCT回路84より供給された画像データが、Bピクチャのデータである場合、可変長復号回路82より供給された予測モードに対応して、フレームメモリ86の前方予測画像部86aに記憶されているIピクチャの画像データ(前方予測モードの場合)、後方予測画像部86bに記憶されているPピクチャの画像データ(後方予測モードの場合)、または、その両方の画像データ(両方向予測モードの場合)が読み出され、動き補償回路87において、可変長復号回路82より出力された動きベクトルに対応する動き補償が施されて、予測画像が生成される。但し、動き補償を必要としない場合(画像内予測モードの場合)、予測画像は生成されない。

【0274】このようにして、動き補償回路87で動き補償が施されたデータは、演算器85において、IDCT回路84の出力と加算される。この加算出力は、フォーマット変換回路32に出力される。

【0275】ただし、この加算出力はBピクチャのデータであり、他の画像の予測画像生成のために利用されることがないため、フレームメモリ86には記憶されない。

【0276】Bピクチャの画像が出力された後、後方予測画像部86bに記憶されているPピクチャの画像データが読み出され、動き補償回路87を介して演算器85に供給される。但し、このとき、動き補償は行われない。

【0277】なお、このデコーダ31には、図5のエンコーダ18における予測モード切り替え回路52とDCTモード切り替え回路55に対応する回路が図示されていないが、これらの回路に対応する処理、すなわち、奇数フィールドと偶数フィールドのラインの信号が分離された構成を元の構成に必要なに応じて戻す処理は、動き補償回路87により実行される。

【0278】また、上述した説明においては、輝度信号の処理について説明したが、色差信号の処理も同様に行われる。ただし、この場合の動きベクトルは、輝度信号用の動きベクトルを、垂直方向および水平方向に1/2にしたものが用いられる。

【0279】図13は、符号化された画像の品質を示している。画像の品質(SNR:Signal to Noise Ratio)は、ピクチャタイプに対応して制御され、Iピクチャ、およびPピクチャは高品質とされ、Bピクチャは、I、Pピクチャに比べて劣る品質とされて伝送される。これは、人間の視覚特性を利用した手法であり、全ての画像品質を平均化するよりも、品質を振動させたほうが視覚上の画質が良くなるためである。このピクチャタイプに対応した画質の制御は、図7の量子化回路57により実行される。

【0280】図14及び図15は、本発明を適用したトランスコーダ101の構成を示しており、図15は、図14のさらに詳細な構成を示している。このトランスコーダ101は、復号装置102に入力された符号化ビデオビットストリーム(encoded video bit stream)のGOP構造及びビットレートを、オペレータの所望するGOP構造及びビットレートに変換する。このトランスコーダ101の機能を説明するために、図15には図示されていないが、このトランスコーダ101の前段に、このトランスコーダ101とほぼ同様の機能を有した3つのトランスコーダが接続されているものとする。つまり、ビットストリームのGOP構造及びビットレートをさまざまに変更するために、第1のトランスコーダ、第2のトランスコーダ、および第3のトランスコーダが順に直列に接続され、その第3のトランスコーダの後ろに、この図1

5に示された第4のトランスコードが接続されているものとする。

【0281】本発明の以下の説明において、この第1のトランスコードにおいて行われた符号化処理を第1世代の符号化処理と定義し、第1のトランスコードの後ろに接続された第2のトランスコードにおいて行われた符号化処理を第2世代の符号化処理と定義し、第2のトランスコードの後ろに接続された第3のトランスコードにおいて行われた符号化処理を第3世代の符号化処理と定義し、第3のトランスコードの後ろに接続された第4のトランスコード（図15に示されたトランスコード101）において行われる符号化処理を第4世代の符号化処理または現在の符号化処理と定義することにする。

【0282】また、第1世代の符号化処理において生成された符号化パラメータを第1世代の符号化パラメータと呼び、第2世代の符号化処理において生成された符号化パラメータを第2世代の符号化パラメータと呼び、第3世代の符号化処理において生成された符号化パラメータを第3世代の符号化パラメータと呼び、第4世代の符号化処理において生成された符号化パラメータを第4世代の符号化パラメータまたは現在の符号化パラメータと呼ぶことにする。

【0283】まず、この図15に示されたトランスコード101に供給される符号化ビデオストリームST (3rd) について説明する。ST (3rd) は、このトランスコード101の前段に設けられている第3のトランスコードにおける第3世代の符号化処理において生成された第3世代の符号化ストリームであることを表わしている。この第3世代の符号化処理において生成された符号化ビデオストリームST (3rd) には、第3の符号化処理において生成された第3世代の符号化パラメータが、この符号化ビデオストリームST (3rd) のシーケンス層、GOP層、ピクチャ層、スライス層、及びマクロブロック層に、sequence_header() 関数、sequence_extension() 関数、group_of_pictures_header() 関数、picture_header() 関数、picture_coding_extension() 関数、picture_data() 関数、slice() 関数、及びmacroblock() 関数として記述されている。このように第3の符号化処理によって生成された第3の符号化ストリームに、第3の符号化処理において使用した第3の符号化パラメータを記述することはMPEG2規格において定義されていることであり、何ら新規性は無い。

【0284】本発明のトランスコード101におけるユニークな点は、この第3の符号化ストリームST (3rd) 中に、第3の符号化パラメータを記述するだけでなく、第1世代及び第2世代の符号化処理において生成された第1世代及び第2世代の符号化パラメータも記述されているという点である。

【0285】具体的には、この第1世代及び第2世代の符号化パラメータは、第3世代の符号化ビデオストリー

ムST (3rd) のピクチャ層のユーザデータエリアに、ヒストリストリームhistory_stream()として記述されている。本発明においては、第3世代の符号化ビデオストリームST (3rd) のピクチャ層のユーザデータエリアに記述されているヒストリストリームを、「ヒストリ情報」、または「履歴情報」と呼び、このヒストリストリームとして記述されている符号化パラメータを「ヒストリパラメータ」、または「履歴パラメータ」と呼んでいる。

【0286】また別の呼び方として、第3世代の符号化ストリームST (3rd) に記述されている第3世代の符号化パラメータを「現在の符号化パラメータ」と呼んだ場合には、第3世代の符号化処理からみて第1世代及び第2世代の符号化処理は、過去に行なわれた符号化処理であるので、第3世代の符号化ストリームST (3rd) のピクチャ層のユーザデータエリアに記述されているヒストリストリームとして記述されている符号化パラメータを「過去の符号化パラメータ」とも呼んでいる。

【0287】このように、この第3の符号化ストリームST (3rd) 中に、第3の符号化パラメータを記述するだけでなく、第1世代及び第2世代の符号化処理において生成された第1世代及び第2世代の符号化パラメータを記述する理由は、トランスコーディング処理によって符号化ストリームのGOP構造やビットレートの変更を繰り返したとしても、画質劣化を防止することができるからである。

【0288】例えば、あるピクチャを第1世代の符号化処理においてPピクチャとして符号化し、第1世代の符号化ストリームのGOP構造を変更するために、第2世代の符号化処理においてそのピクチャをBピクチャとして符号化し、第2世代の符号化ストリームのGOP構造をさらに変更するために、第3世代の符号化処理において、再度そのピクチャをPピクチャとして符号化することが考えられる。MPEG規格に基づく符号化処理及び復号処理は100%可逆の処理ではないので、符号化及び復号処理を繰り返す毎に画質が劣化していくことは知られている。

【0289】このような場合に、第3の世代の符号化処理において、量子化スケール、動きベクトル、予測モードなどの符号化パラメータをもう一度計算するのではなくて、第1世代の符号化処理において生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータを再利用する。第3世代の符号化処理によって新しく生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータよりも、第1世代の符号化処理によって新しく生成された量子化スケール、動きベクトル、予測モードなどの符号化パラメータの方が、明らかに精度が良いので、この第1世代のパラメータを再利用することによって、符号化及び復号処理を繰り返したとしても画質劣化を少なくすることができる。

【0290】上述した本発明にかかる処理を説明するために、図15に示された第4世代のトランスコーダ101の処理を例に挙げてより詳しく説明する。

【0291】復号装置102は、第3世代の符号化ビットストリームST (3rd)に含まれている符号化ビデオを第3世代の符号化パラメータを使用して復号し、復号されたベースバンドのデジタルビデオデータを生成するための装置である。さらに、復号装置102は、第3世代の符号化ビットストリームST (3rd)のピクチャ層のユーザデータエリアにヒストリストリームとして記述されている第1世代及び第2世代の符号化パラメータをデコードするための装置でもある。

【0292】具体的には、図16に示されているように、復号装置102は、図5の復号装置2のデコーダ31 (図12)と基本的に同様の構成とされ、供給されたビットストリームをバッファリングするための受信バッファ81、符号化ビットストリームを可変長復号するための可変長復号回路112、可変長復号されたデータを可変長復号回路112から供給された量子化スケールに従って逆量子化する逆量子化回路83、逆量子化されたDCT係数を逆離散コサイン変換するIDCT回路84、及び動き補償処理を行うための演算器85、フレームメモリ86及び動き補償回路87を備えている。

【0293】可変長復号回路112は、第3世代の符号化ビットストリームST (3rd)を復号処理するために、この第3世代の符号化ビットストリームST (3rd)のピクチャ層、スライス層及びマクロブロック層に記述されている第3世代の符号化パラメータを抽出する。たとえば、この可変長復号回路112において抽出される第3世代の符号化パラメータは、ピクチャタイプを示すpicture_coding_type、量子化スケールステップサイズを示すquantiser_scale_code、予測モードを示すmacroblock_type、動きベクトルを示すmotion_vector、Frame予測モードかField予測モードを示すframe/field_motion_type、及びFrameDCTモードかFieldDCTモードを示すdct_type等である。この可変長復号回路112において抽出されたquantiser_scale_codeは、逆量子化回路83に供給され、picture_coding_type、quantiser_scale_code、macroblock_type、motion_vector、frame/field_motion_type、dct_type等のパラメータは、動き補償回路87に供給される。

【0294】可変長復号回路112は、第3世代の符号化ビットストリームST (3rd)を復号処理するために必要なこれらの符号化パラメータだけではなく、後段の第5世代のトランスコーダに第3世代のヒストリ情報として伝送されるべき符号化パラメータを、第3世代の符号化ビットストリームST (3rd)のシーケンス層、GOP層、ピクチャ層、スライス層、及びマクロブロック層から抽出する。もちろん、第3世代の復号処理に使用されたpicture_coding_type、quantiser_scale_code、macr

oblock_type、motion_vector、frame/field_motion_type、dct_type等の第3世代の符号化パラメータは、この第3世代のヒストリ情報に含まれている。ヒストリ情報としてどのような符号化パラメータを抽出するかについては、伝送容量などに応じてオペレータやホストコンピュータ側からあらかじめ設定されている。

【0295】さらに、可変長復号回路112は、第3世代の符号化ビットストリームST (3rd)のピクチャ層のユーザデータエリアに記述されているユーザデータを抽出し、そのユーザデータをヒストリデコーディング装置104に供給する。

【0296】このヒストリデコーディング装置104は、第3世代の符号化ビットストリームST (3rd)のピクチャ層に記述されていたユーザデータから、ヒストリ情報として記述されている第1世代の符号化パラメータ及び第2世代の符号化パラメータ (直前の世代よりさらに前の世代の符号化パラメータ)を抽出するための回路である。具体的には、ヒストリデコーディング装置104は、受け取ったユーザデータのシンタックスを解析することによって、ユーザデータの中に記述されている固有のHistory_Data_Idを検出し、これによって、converted_history_stream()を抽出することができる。さらに、ヒストリデコーディング装置104は、converted_history_stream()中にある所定間隔に挿入されている1ビットのマーカービット(marker_bit)を取りさることによって、history_stream()を得、そして、そのhistory_stream()のシンタックスを解析することによって、history_stream()中に記述されている第1世代及び第2世代の符号化パラメータを得ることができる。このヒストリデコーディング装置104の詳しい動作については、後述する。

【0297】ヒストリ情報多重化装置103は、第1世代、第2世代及び第3世代の符号化パラメータを、第4世代の符号化処理を行う符号化装置106に供給するために、復号装置102においてデコードされたベースバンドのビデオデータに、これらの第1世代、第2世代及び第3世代の符号化パラメータを多重化するための回路である。具体的には、ヒストリ情報多重化装置103は、復号装置102の演算器85から出力されたベースバンドのビデオデータ、復号装置102の可変長復号装置112から出力された第3世代の符号化パラメータ、並びに、ヒストリデコーディング装置104から出力された第1世代の符号化パラメータと第2世代の符号化パラメータとを受け取り、このベースバンドのビデオデータに、これらの第1世代、第2世代及び第3世代の符号化パラメータを多重化する。第1世代、第2世代及び第3世代の符号化パラメータが多重化されたベースバンドのビデオデータは、伝送ケーブルを介してヒストリ情報分離装置105に供給される。

【0298】次に、これらの第1世代、第2世代及び第

3世代の符号化パラメータのベースバンドビデオデータへの多重化の方法について、図17及び図18を参照して説明する。、図17は、MPEG規格において定義されている、16ピクセル×16ピクセルからなる1つのマクロブロックを示している。この16ピクセル×16ピクセルのマクロブロックは、輝度信号に関しては4つの8ピクセル×8ピクセルからなるサブブロック (Y[0], [1], [2]及びY[3]) と、色差信号に関しては4つの8ピクセル×8ピクセルからなるサブブロック (Cr[0], r[1], b[0], 及びCb[1]) から構成されている。

【0299】図18は、ビデオデータのあるフォーマットを表している。このフォーマットは、ITU勧告-RDT60.1において定義されているフォーマットであって、放送業界において使用されている所謂「D1フォーマット」をあらわしている。このD1フォーマットは、10ビットのビデオデータを伝送するためのフォーマットとして規格化されたので、ビデオデータの1ピクセルを10ビットで表現できるようになっている。

【0300】MPEG規格によってデコードされたベースバンドのビデオデータは8ビットであるので、本発明のトランスコーダにおいては、図18に示したように、D1フォーマットの10ビットのうち上位8ビット (D9乃至D2) を使用して、MPEG規格にもとづいてデコードされたベースバンドのビデオデータを伝送するようにしている。このように、復号された8ビットのビデオデータをD1フォーマットに書き込むと、下位2ビット (D1とD0) は、空きビット (unallocated bits) となる。本発明のトランスコーダではこの空きエリア (unallocated area) を利用して、ヒストリ情報を伝送するようにしている。

【0301】この図18に記載されたデータブロックは、各サブブロック (Y[0], Y[1], Y[2], Y[3], Cr[0], Cr[1], Cb[0], Cb[1]) における1ピクセルを伝送するためのデータブロックであるので、1マクロブロックのデータを伝送するためには、この図18に示されているデータブロックが64個伝送される。下位2ビット (D1とD0) を使用すれば、1マクロブロックのビデオデータに対して、合計で1024 (=16×64) ビットのヒストリ情報を伝送できる。従って、1世代分のヒストリ情報は、256ビットとなるように生成されているので、過去の4 (=1024/256) 世代分のヒストリ情報を1マクロブロックのビデオデータに対して重畳することができる。図18に示した例では、第1世代のヒストリ情報、第2世代のヒストリ情報、並びに、第3世代のヒストリ情報が重畳されている。

【0302】ヒストリ情報分離装置105は、D1フォーマットとして伝送されたデータの上位8ビットから、ベースバンドビデオデータを抽出し、下位2ビットから

ヒストリ情報を抽出するための回路である。図15に示した例では、ヒストリ情報分離装置105は、伝送データからベースバンドのビデオデータを抽出して、そのビデオデータを符号化装置106に供給するとともに、伝送データから第1世代、第2世代及び第3世代のヒストリ情報を抽出して、符号化装置106とヒストリエncoding装置107にそれぞれ供給する。

【0303】符号化装置106は、ヒストリ情報分離装置105から供給されたベースバンドのビデオデータを、オペレータまたはホストコンピュータから指定されたGOP構造及びビットレートを有するビットストリームになるように符号化するための装置である。なお、GOP構造を変更するとは、たとえば、GOPに含まれるピクチャの数、IピクチャとIピクチャの間に存在するPピクチャの数、及びIピクチャとPピクチャ (またはIピクチャ) の間に存在するBピクチャの数を変更することを意味する。

【0304】図15に示された例では、供給されたベースバンドのビデオデータには、第1世代、第2世代及び第3世代のヒストリ情報が重畳されているので、この符号化装置106は、再符号化処理による画質劣化が少なくなるように、これらのヒストリ情報を選択的に再利用して第4世代の符号化処理を行う。

【0305】図19は、この符号化装置106に設けられているエンコーダ121の具体的な構成を示している図である。このエンコーダ121は、基本的には、図7に示したエンコーダ18と同様に構成され、動きベクトル検出回路50、フレーム/フィールド予測モード切り替え回路52、演算器53、DCTモード切り替え回路55、DCT回路56、量子化回路57、可変長符号化回路58、伝送バッファ59、逆量子化回路60、逆DCT回路61、演算器62、フレームメモリ63、並びに動き補償回路64を備えている。これらの、各回路の機能は、図7において説明したエンコーダ18における場合の機能とほぼ同じであるので、その説明は省略する。以下に、このエンコーダ121と、図7において説明したエンコーダ18との異なる点を中心に説明する。

【0306】このエンコーダ121は、上述した各回路の動作及び機能を制御するためのコントローラ70を有している。このコントローラ70は、オペレータまたはホストコンピュータからGOP構造に関するインストラクションを受け取って、そのGOP構造に対応するように各ピクチャのピクチャタイプを決定する。また、このコントローラ70は、オペレータまたはホストコンピュータからターゲットビットレートの情報を受け取り、このエンコーダ121から出力されるビットレートがこの指定されたターゲットビットレートになるように、量子化回路57を制御する。

【0307】さらに、このコントローラ70は、ヒストリ情報分離装置105から出力された複数世代のヒスト

り情報を受け取り、これらのヒストリ情報を再利用して参照ピクチャの符号化処理を行う。以下に詳しく説明する。

【0308】まず、このコントローラ70は、オペレータによって指定されたGOP構造から決定された参照ピクチャのピクチャタイプと、ヒストリ情報に含まれるピクチャタイプが一致するか否かを判断する。つまり、指定されたピクチャタイプと同じピクチャタイプでこの参照ピクチャが過去において符号化されたことがあるか否かを判断する。

【0309】図15に示された例をあげてよりわかりやすく説明するのであれば、このコントローラ70は、第4世代の符号化処理としてこの参照ピクチャにアサインされたピクチャタイプが、第1世代の符号化処理におけるこの参照ピクチャのピクチャタイプ、第2世代の符号化処理におけるこの参照ピクチャのピクチャタイプ、または第3世代の符号化処理におけるこの参照ピクチャのピクチャタイプのいずれかと一致するか否かを判断する。

【0310】もし、第4世代の符号化処理としてこの参照ピクチャに指定されたピクチャタイプが、過去の符号化処理におけるどのピクチャタイプとも一致しないのであれば、このコントローラ70は、「通常符号化処理」を行う。つまり、この場合には、第1世代、第2世代または第3世代のどの世代の符号化処理においても、第4世代の符号化処理としてアサインされたピクチャタイプで、この参照ピクチャが符号化処理されたことがないということになる。一方、もし、第4世代の符号化処理としてこの参照ピクチャに指定されたピクチャタイプが、過去の符号化処理におけるいずれかのピクチャタイプと一致するのであれば、このコントローラ70は、「パラメータ再利用符号化処理」を行う。つまり、この場合には、第1世代、第2世代または第3世代のいずれかの世代の符号化処理において、第4世代の符号化処理としてアサインされたピクチャタイプで、この参照ピクチャが符号化処理されたことがあるということになる。

【0311】まず、最初にコントローラ70の通常符号化処理について説明する。

【0312】動きベクトル検出回路50は、フレーム予測モードまたはフィールド予測モードのどちらが選択されるべきかを判断するために、フレーム予測モードにおける予測誤差とフィールド予測モードにおける予測誤差をそれぞれ検出し、その予測誤差の値をコントローラ70に供給する。コントローラ70は、それらの予測誤差の値を比較し、その予測誤差の値が小さい方の予測モードを選択する。予測モード切り替え回路52は、コントローラ70によって選択された予測モードに対応するように信号処理を行い、それを演算器53に供給する。

【0313】具体的には、予測モード切り替え回路52は、フレーム予測モードが選択された場合には、図8を参照して説明したように、輝度信号に関しては、入力さ

れた状態のまま演算器53に出力するように信号処理を行い、色差信号に関しては、奇数フィールドラインと偶数フィールドラインとが混在するように信号処理する。一方、フィールド予測モードが選択された場合には、図9を参照して説明したように、輝度信号に関しては、輝度ブロックY[1]とY[2]を奇数フィールドラインで構成し、輝度ブロックY[3]とY[4]を偶数フィールドラインで構成するように信号処理し、色差信号に関しては、上4ラインを奇数フィールドラインで構成し、下4ラインを偶数フィールドラインで構成するように信号処理する。

【0314】さらに、動きベクトル検出回路50は、画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのうちのいずれの予測モードを選択するかを決定するために、各予測モードにおける予測誤差を生成し、各予測モードにおける予測誤差をコントローラ70にそれぞれ供給する。コントローラ70は、前方予測、後方予測および両方向予測の予測誤差のうちの最も小さいものを、インタ予測の予測誤差として選択する。さらに、このインタ予測の予測誤差と、画像内予測の予測誤差とを比較し、その小さい方を選択し、この選択した予測誤差に対応するモードを予測モードとして選択する。すなわち、画像内予測の予測誤差の方が小さければ、画像内予測モードが設定される。インタ予測の予測誤差の方が小さければ、前方予測、後方予測または両方向予測モードのうちの対応する予測誤差が最も小さかったモードが設定される。コントローラ70は、選択した予測モードに対応するように、演算器53及び動き補償回路64を制御する。

【0315】DCTモード切り替え回路55は、フレームDCTモードまたはフィールドDCTモードのいずれかを選択するために、4個の輝度ブロックのデータを、奇数フィールドラインと偶数フィールドラインが混在するような信号形態（フレームDCTモード）に変換するとともに、奇数フィールドラインと偶数フィールドラインが分離された信号形態（フィールドDCTモード）に変換して、それぞれの信号をDCT回路56に供給する。DCT回路56は、奇数フィールドと偶数フィールドを混在してDCT処理した場合における符号化効率と、奇数フィールドと偶数フィールドを分離した状態においてDCT処理した場合の符号化効率を計算し、その結果をコントローラ70に供給する。コントローラ70は、DCT回路56から供給されたそれぞれの符号化効率を比較し、符号化効率の良い方のDCTモードを選択し、その選択したDCTモードとなるようにDCTモード切り替え回路55を制御する。

【0316】コントローラ70は、オペレータまたはホストコンピュータから供給された目標ビットレートを示すターゲットビットレートと、送信バッファ59にバッファリングされているビット量を示す信号、つまり、パ

ッファ残量を示す信号を受け取り、このターゲットビットレートとバッファ残量に基づいて、量子化回路 57 の量子化ステップサイズをコントロールするための feedback_q_scale_code を生成する。この feedback_q_scale_code は、この送信バッファ 59 がオーバーフローまたはアンダーフローしないように、この送信バッファ 59 のバッファ残量に応じて生成される制御信号であって、また、送信バッファ 59 から出力されるビットストリームのビットレートが、ターゲットビットレートになるように制御する信号でもある。

【0317】具体的には、例えば、送信バッファ 59 にバッファリングされているビット量が少なくなってしまう場合には、次に符号化するピクチャの発生ビット量が増えるように、量子化ステップサイズを小さくし、一方、送信バッファ 59 にバッファリングされているビット量が多くなってしまった場合には、次に符号化するピクチャの発生ビット量が少なくなるように、量子化ステップサイズを大きくする。なお、feedback_q_scale_code と量子化ステップサイズは比例し、feedback_q_scale_code を大きくすると、量子化ステップサイズは大きくなり、feedback_q_scale_code を小さくすると、量子化ステップサイズは小さくなる。

【0318】次に、このトランスコード 101 の特徴の 1 つでもある、パラメータ再利用符号化処理について説明する。この処理をより分かりやすく説明するために、参照ピクチャは、第 1 世代の符号化処理において P ピクチャとして符号化され、第 2 世代の符号化処理において I ピクチャとして符号化処理され、第 3 世代の符号化処理において B ピクチャとして符号化されていたものとし、今回の第 4 世代の符号化処理において、この参照ピクチャを P ピクチャとして符号化しなければいけないものとする。

【0319】この場合には、第 4 世代のピクチャタイプとしてアサインされたピクチャタイプと同じピクチャタイプ（I ピクチャ）で、この参照ピクチャは第 1 世代の符号化処理において符号化されているので、コントローラ 70 は、供給されたビデオデータから符号化パラメータを新しく作成するのではなく、第 1 世代の符号化パラメータを使用して符号化処理を行う。この第 4 の符号化処理において再利用する符号化パラメータは、代表的なパラメータとしては、量子化スケールステップサイズを示す quantiser_scale_code、予測方向モードを示す macroblock_type、動きベクトルを示す motion_vector、Frame 予測モードか Field 予測モードを示す frame/field_motion_type、及び Frame DCT モードか Field DCT モードを示す dct_type 等である。

【0320】コントローラ 70 は、履歴情報として伝送されたすべての符号化パラメータを再利用するわけではなく、再利用した方が望ましいと想定される上述したような符号化パラメータについては再利用し、再利用

しない方が望ましいと考えられる符号化パラメータについては、新しく生成する。

【0321】次に、この符号化パラメータ再利用符号化処理について、上述した通常符号化処理と異なる点を中心に説明する。

【0322】動きベクトル検出回路 50 は、上述した通常符号化処理においては、参照ピクチャの動きベクトルの検出を行ったが、このパラメータ再利用符号化処理においては、動きベクトル motion_vector の検出処理は行わずに、第 1 世代の履歴情報として供給された動きベクトル motion_vector を再利用する。その理由について説明する。

【0323】第 3 世代の符号化ストリームを復号したベースバンドのビデオデータは、少なくとも 3 回の復号及び符号化処理が行われているので、オリジナルビデオデータに比べると、明らかに画質が劣化している。画質が劣化しているビデオデータから動きベクトルを検出したとしても、正確な動きベクトルは検出できない。つまり、第 4 世代の符号化処理において検出された動きベクトルよりも、第 1 世代の履歴情報として供給されている動きベクトルの方が、明らかに、精度の高い動きベクトルである。つまり、第 1 世代の符号化パラメータとして伝送された動きベクトルを再利用することによって、第 4 世代の符号化処理を行ったとしても画質が劣化しない。コントローラ 70 は、この第 1 世代の履歴情報として供給された動きベクトル motion_vector を、第 4 世代の符号化処理において符号化されるこの参照ピクチャの動きベクトル情報として、動き補償回路 64 及び可変長符号化回路 58 に供給する。

【0324】さらに、動きベクトル検出回路 50 は、フレーム予測モードとフィールド予測モードのどちらが選択されるかを判断するために、フレーム予測モードにおける予測誤差とフィールド予測モードにおける予測誤差をそれぞれ検出したが、このパラメータ再利用符号化処理においては、このフレーム予測モードにおける予測誤差とフィールド予測モードにおける予測誤差を検出する処理は行わずに、第 1 世代の履歴情報として供給されている Frame 予測モードか Field 予測モードを示す frame/field_motion_type を再利用する。なぜなら、第 4 世代の符号化処理において検出された各予測モードにおける予測誤差よりも、第 1 世代において検出された各予測モードにおける予測誤差の方が精度が高いため、精度の高い予測誤差によって決定された予測モードを選択した方がより最適な符号化処理が行うことができるからである。

【0325】具体的には、コントローラ 70 は、この第 1 世代の履歴情報として供給されている frame/field_motion_type に対応する制御信号を予測モード切り替え回路 52 に供給し、予測モード切り替え回路 52 は、この再利用された frame/field_motion_type に対応した

信号処理を行う。

【0326】さらには、動きベクトル検出回路50は、通常符号化処理においては、画像内予測モード、前方予測モード、後方予測モード、または両方向予測モードのうちのいずれの予測モード（以下、この予測モードを、予測方向モードとも称する）を選択するかを決定するために、各予測方向モードにおける予測誤差を計算していたが、このパラメータ再利用符号化処理においては、各予測方向モードにおける予測誤差の計算は行わず、第1世代のヒストリ情報として供給されたmacroblock_typeに基づいて予測方向モードを決定する。なぜなら、第4世代の符号化処理における各予測方向モードにおける予測誤差よりも、第1世代の符号化処理における各予測方向モードにおける予測誤差の方がより精度が高いため、より精度の高い予測誤差によって決定された予測方向モードを選択した方が、より効率の良い符号化処理が行えるからである。具体的には、コントローラ70は、第1世代のヒストリ情報に含まれているmacroblock_typeによって示される予測方向モードを選択し、その選択した予測方向モードに対応するように、演算器53及び動き補償回路64をコントロールする。

【0327】DCTモード切り替え回路55は、通常符号化処理においては、フレームDCTモードの符号化効率と、フィールドDCTモードの符号化効率を比較するために、フレームDCTモードの信号形態に変換した信号と、フィールドDCTモードの信号形態に変換した信号の両方をDCT回路56に供給していたが、このパラメータ再利用符号化処理では、フレームDCTモードの信号形態に変換した信号と、フィールドDCTモードの信号形態に変換した信号の両方を生成する処理は行っておらず、第1世代のヒストリ情報に含まれているdct_typeによって示されたDCTモードに対応する処理のみを行っている。具体的には、コントローラ70は、第1世代のヒストリ情報に含まれているdct_typeを再利用し、DCTモード切り替え回路55がこのdct_typeによって示されるDCTモードに対応した信号処理を行うように、DCTモード切り替え回路55をコントロールする。

【0328】コントローラ70は、通常符号化処理では、オペレータによって指定されたターゲットビットレートと送信バッファ残量に基づいて、量子化回路57の量子化ステップサイズをコントロールしていたが、このパラメータ再利用符号化処理では、ターゲットビットレート、送信バッファ残量及びヒストリ情報に含まれている過去の量子化スケールに基づいて、量子化回路57の量子化ステップサイズをコントロールする。なお、以下の説明において、ヒストリ情報に含まれている過去の量子化スケールをhistory_q_scale_codeと記述することにする。また、後述するヒストリストリームにおいては、この量子化スケールを、quantiser_scale_codeと記述している。

【0329】まず、コントローラ70は、通常符号化処理と同じように、現在の量子化スケールfeedback_q_scale_codeを生成する。このfeedback_q_scale_codeは、この送信バッファ59がオーバーフロー及びアンダーフローしないように、この送信バッファ59のバッファ残量に応じて決定される値である。続いて、第1世代のヒストリストリームに含まれている過去の量子化スケールhistory_q_scale_codeの値と、この現在の量子化スケールfeedback_q_scale_codeの値を比較し、どちらの量子化スケールの方が大きいかを判断する。量子化スケールが大きいとは、量子化ステップが大きいことを意味する。もし、現在の量子化スケールfeedback_q_scale_codeが、過去の量子化スケールhistory_q_scale_codeよりも大きいのであれば、コントローラ70は、この現在の量子化スケールfeedback_q_scale_codeを量子化回路57に供給する。一方、過去の量子化スケールhistory_q_scale_codeが、現在の量子化スケールfeedback_q_scale_codeよりも大きいのであれば、コントローラ70は、この過去の量子化スケールhistory_q_scale_codeを量子化回路57に供給する。

【0330】つまり、コントローラ70は、ヒストリ情報に含まれている複数の過去の量子化スケールと、送信バッファの残量から計算された現在の量子化スケールの中で、もっとも大きい量子化スケールコードを選択する。また、別の言葉で説明するのであれば、コントローラ70は、過去（第1、第2及び第3世代）の符号化処理における量子化ステップまたは現在（第4世代）の符号化処理において使用された量子化ステップの中で、もっとも大きい量子化ステップを使用して量子化を行うように量子化回路57を制御する。この理由を以下に説明する。

【0331】たとえば、第3世代の符号化処理において生成されたストリームのビットレートが4[Mbps]であって、この第4世代の符号化処理を行うエンコーダ121に対して設定されたターゲットビットレートが15[Mbps]であったとする。このときに、ターゲットビットレートが上がっているので、単純に量子化ステップを小さくすれば良いかという、実際にはそうではない。過去の符号化処理において大きい量子化ステップで符号化処理されたピクチャを、現在の符号化処理において、量子化ステップを小さくして符号化処理を行ったとしても、このピクチャの画質は向上することはない。つまり、過去の符号化処理における量子化ステップよりも小さい量子化ステップで符号化することは、単にビット量が増えるだけであって、画質を向上させることにはならない。よって、過去（第1、第2及び第3世代）の符号化処理における量子化ステップまたは現在（第4世代）の符号化処理において使用された量子化ステップの中で、もっとも大きい量子化ステップを使用して量子化を行うと、もっとも効率の良い符号化処理が行える。

【0332】次に、図15におけるヒストリデコーディング装置104とヒストリエンコーディング装置107についてさらに説明する。同図に示すように、ヒストリデコーディング装置104は、復号装置102より供給されるユーザデータをデコードするユーザデータデコーダ201、ユーザデータデコーダ201の出力を変換するコンバータ202、およびコンバータ202の出力から履歴情報を再生するヒストリVLD203により構成されている。

【0333】また、ヒストリエンコーディング装置107は、ヒストリ情報分離装置105より供給される3世代分の符号化パラメータをフォーマット化するヒストリVLC211、ヒストリVLC211の出力を変換するコンバータ212、コンバータ212の出力をユーザデータのフォーマットにフォーマットするユーザデータフォーマッタ213により構成されている。

【0334】ユーザデータデコーダ201は、復号装置102より供給されるユーザデータをデコードして、コンバータ202に出力する。詳細は図51を参照して後述するが、ユーザデータ (user_data0) は、user_data_start_codeとuser_dataからなり、MPEG規格においてはuser_dataの中に、連続する23ビットの"0" (start_codeと同一のコード) を発生させることを禁止している。これは、そのデータが、start_codeとして誤検出されるのを防止するためである。履歴情報 (history_stream0) は、ユーザデータエリアに (MPEG規格のuser_dataの一種として) 記述され、その中には、このような連続する23ビット以上の"0"が存在することがあり得るので、これを、連続する23ビット以上の"0"が発生しないように、所定のタイミングで"1"を挿入処理して、converted_history_stream0 (後述する図38) に変換する必要がある。この変換を行うのは、ヒストリエンコーディング装置107のコンバータ212である。ヒストリデコーディング装置104のコンバータ202は、このコンバータ212と逆の変換処理を行う (連続する23ビット以上の"0"を発生させないために挿入された"1"を除去する) ものである。

【0335】ヒストリVLD203は、コンバータ202の出力から履歴情報 (いまの場合、第1世代の符号化パラメータと第2世代の符号化パラメータ) を生成し、ヒストリ情報多重化装置103に出力する。

【0336】一方、ヒストリエンコーディング装置107においては、ヒストリVLC211がヒストリ情報分離装置105より供給される3世代分の (第1世代、第2世代、および第3世代の) 符号化パラメータを履歴情報のフォーマットに変換する。このフォーマットには、固定長のもの (後述する図40乃至図46) と、可変長のもの (後述する図47) とがある。これらの詳細については後述する。

【0337】ヒストリVLC211により、フォーマット

化された履歴情報は、コンバータ212において、converted_history_stream0に変換される。これは、上述したように、user_data0のstart_codeが誤検出されないようにするための処理である。すなわち、履歴情報内には連続する23ビット以上の"0"が存在するが、user_data中には連続する23ビット以上の"0"を配置することができないので、この禁止項目に触れないようにコンバータ212によりデータを変換する ("1"を所定のタイミングで挿入する) のである。

【0338】ユーザデータフォーマッタ213は、コンバータ212より供給されるconverted_history_stream0に、後述する図38に基づいて、History_Data_IDを付加し、さらに、user_data_stream_codeを付加して、video stream中に挿入できるMPEG規格のuser_dataを生成し、符号化装置106に出力する。

【0339】図20は、ヒストリVLC211の構成例を表している。その符号語変換器301と符号長変換器305には、符号化パラメータ (今回、履歴情報として伝送する符号化パラメータ) (項目データ) と、この符号化パラメータを配置するストリームを特定する情報 (例えば、シンタックスの名称 (例えば、後述するsequence_headerの名称)) (項目NO.) が、ヒストリ情報分離装置105から供給されている。符号語変換器301は、入力された符号化パラメータを、指示されたシンタックスに対応する符号語に変換し、バレルシフト302に出力する。バレルシフト302は、符号語変換器301より入力された符号語を、アドレス発生回路306より供給されるシフト量に対応する分だけシフトし、バイト単位の符号語として、スイッチ303に出力する。アドレス発生回路306が出力するビットセレクト信号により切り換えられるスイッチ303は、ビット分設けられており、バレルシフト302より供給される符号語を、RAM304に供給し、記憶させる。このときの書き込みアドレスは、アドレス発生回路306から指定される。また、アドレス発生回路306から読み出しアドレスが指定されたとき、RAM304に記憶されているデータ (符号語) が読み出され、後段のコンバータ212に供給されるとともに、必要に応じて、スイッチ303を介してRAM304に再び供給され、記憶される。

【0340】符号長変換器305は、入力されるシンタックスと符号化パラメータとから、その符号化パラメータの符号長を決定し、アドレス発生回路306に出力する。アドレス発生回路306は、入力された符号長に対応して、上述したシフト量、ビットセレクト信号、書き込みアドレス、または読み出しアドレスを生成し、それらを、それぞれバレルシフト302、スイッチ303、またはRAM304に供給する。

【0341】以上のように、ヒストリVLC211は、いわゆる可変長符号化器として構成され、入力された符号化パラメータを可変長符号化して出力する。

【0342】図21は、以上のようにしてヒストリフォーマット化されたデータをデコードするヒストリVLD203の構成例を表している。このヒストリVLD203には、コンバータ202から供給された符号化パラメータのデータがRAM311に供給されて、記憶される。このときの書き込みアドレスは、アドレス発生回路315から供給される。アドレス発生回路315はまた、所定のタイミングで読み出しアドレスを発生し、RAM311に供給する。このとき、RAM311は、読み出しアドレスに記憶されているデータを読み出し、バレルシフト312に出力する。バレルシフト312は、アドレス発生回路315が出力するシフト量に対応する分だけ、入力されるデータをシフトし、逆符号長変換器313と逆符号語変換器314に出力する。

【0343】逆符号長変換器313にはまた、コンバータ202から、符号化パラメータが配置されているストリームのシンタックスの名称(項目NO.)が供給されている。逆符号長変換器313は、そのシンタックスに基づいて、入力されたデータ(符号語)から符号長を求め、求めた符号長をアドレス発生回路315に出力する。

【0344】また、逆符号語変換器314は、バレルシフト312より供給されたデータを、シンタックスに基づいて復号し(逆符号語化し)、ヒストリ情報多重化装置103に出力する。

【0345】また、逆符号語変換器314は、どのような符号語が含まれているのかを特定するのに必要な情報(符号語の区切りを決定するのに必要な情報)を抽出し、アドレス発生回路315に出力する。アドレス発生回路315は、この情報と逆符号長変換器313より入力された符号長に基づいて、書き込みアドレスおよび読み出しアドレスを発生し、RAM311に出力するとともに、シフト量を発生し、バレルシフト312に出力する。

【0346】図22は、コンバータ212の構成例を表している。この例においては、ヒストリVLC211とコンバータ212の間に配置されているバッファメモリ320の、コントローラ326が出力する読み出しアドレスから8ビットのデータが読み出され、D型フリップフロップ(D-FF)321に供給され、保持されるようになされている。そして、D型フリップフロップ321より読み出されたデータは、スタンプ回路323に供給されるとともに、8ビットのD型フリップフロップ322にも供給され、保持される。D型フリップフロップ322より読み出された8ビットのデータは、D型フリップフロップ321より読み出された8ビットのデータと合成され、16ビットのバラレルデータとして、スタンプ回路323に供給される。

【0347】スタンプ回路323は、コントローラ326より供給されるスタンプ位置を示す信号(stuff posi

tion)の位置に符号"1"を挿入し(スタフピングし)、合計17ビットのデータとして、バレルシフト324に出力する。

【0348】バレルシフト324は、コントローラ326より供給されるシフト量を示す信号(shift)に基づいて入力されたデータをシフトして、8ビットのデータを抽出し、8ビットのD型フリップフロップ325に出力する。D型フリップフロップ325に保持されたデータは、そこから読み出され、バッファメモリ327を介して、後段のユーザデータフォーマッタ213に供給される。この時、コントローラ326は、出力するデータとともに、書き込みアドレスを発生し、コンバータ212とユーザデータフォーマッタ213との間に介在するバッファメモリ327に供給する。

【0349】図23は、スタンプ回路323の構成例を表している。D型フリップフロップ322、321より入力された16ビットのデータは、それぞれスイッチ331-16乃至331-1の接点aに入力されている。スイッチ331-i(i=0乃至15)の接点cには、MSB側(図中上方)に隣接するスイッチのデータが供給されている。例えば、スイッチ331-12の接点cには、MSB側に隣接するスイッチ331-13の接点aに供給されているLSBから13番目のデータが供給されており、スイッチ331-13の接点cには、MSB側に隣接するスイッチ331-14の接点aに供給されているLSB側から14番目のデータが供給されている。

【0350】但し、LSBに対応するスイッチ331-1よりさらに下側のスイッチ331-0の接点aは、開放されている。また、MSBに対応するスイッチ331-16の接点cは、それより上位のスイッチが存在しないため、開放されている。

【0351】各スイッチ331-0乃至331-16の接点bには、データ"1"が供給されている。

【0352】デコーダ332は、コントローラ326より供給されるデータ"1"を挿入する位置を示す信号stuff positionに対応して、スイッチ331-0乃至331-16のうち、1つのスイッチを接点b側に切り替え、それよりLSB側のスイッチは、接点c側にそれぞれ切り替えさせ、それよりMSB側のスイッチは、接点a側に切り替えさせる。

【0353】図23は、LSB側から13番目にデータ"1"を挿入する場合の例を示している。従って、この場合、スイッチ331-0乃至スイッチ331-12は、いずれも接点c側に切り替えられ、スイッチ331-13は、接点b側に切り替えられ、スイッチ331-14乃至スイッチ331-16は、接点a側に切り替えられている。

【0354】図22のコンバータ212は、以上のような構成により、22ビットの符号を23ビットに変換して、出力することになる。

【0355】図24は、図22のコンバータ212の各部の出力データのタイミングを表している。コンバータ212のコントローラ326がバイト単位のクロックに同期して、読み出しアドレス（図24（A））を発生すると、バッファメモリ320から、それに対応するデータが、バイト単位で読み出され、D型フリップフロップ321に一旦保持される。そして、D型フリップフロップ321より読み出されたデータ（図24（B））は、スタッフ回路323に供給されるとともに、D型フリップフロップ322に供給され、保持される。D型フリップフロップ322に保持されたデータは、そこからさらに読み出され（図24（C））、スタッフ回路323に供給される。

【0356】従って、スタッフ回路323の入力（図24（D））は、読み出しアドレスA1のタイミングにおいて、最初の1バイトのデータD0とされ、次の読み出しアドレスA2のタイミングにおいて、1バイトのデータD0と1バイトのデータD1より構成される2バイトのデータとなり、さらに読み出しアドレスA3のタイミングにおいては、データD1とデータD2より構成される2バイトのデータとなる。

【0357】スタッフ回路323には、データ"1"を挿入する位置を示す信号stuff position（図24（E））がコントローラ326より供給される。スタッフ回路323のデコーダ332は、スイッチ331-16乃至331-0のうち、この信号stuff positionに対応するスイッチを接点bに切り換え、それよりLSB側のスイッチを接点c側に切り換え、さらにそれよりMSB側のスイッチを接点a側に切り換える。これにより、データ"1"が挿入されるので、スタッフ回路323からは、信号stuff positionで示す位置に、データ"1"が挿入されたデータ（図24（F））が出力される。

【0358】バレルシフタ324は、入力されたデータを、コントローラ326より供給される信号shift（図24（G））で示される量だけバレルシフトして、出力する（図24（H））。この出力がさらにD型フリップフロップ325で一旦保持された後、後段に出力される（図24（I））。

【0359】D型フリップフロップ325より出力されるデータには、22ビットのデータの次に、データ"1"が挿入されている。従って、データ"1"と、次のデータ"1"の間には、その間のビットが全て0であったとしても、0のデータの連続する数は22となる。

【0360】図25は、コンバータ202の構成例を表している。このコンバータ202のD型フリップフロップ341乃至コントローラ346よりなる構成は、図22に示したコンバータ212のD型フリップフロップ321乃至コントローラ326と基本的に同様の構成であるが、コンバータ212におけるスタッフ回路323に代えて、ディリート回路343が挿入されている点がコ

ンバータ212における場合と異なっている。その他の構成は、図22のコンバータ212における場合と同様である。

【0361】すなわち、このコンバータ202においては、コントローラ346が出力する削除するビットの位置を示す信号delete positionに従って、ディリート回路343が、そのビット（図22のスタッフ回路323で挿入されたデータ"1"）が削除される。

【0362】その他の動作は、図22のコンバータ212における場合と同様である。

【0363】図26は、ディリート回路343の構成例を表している。この構成例においては、D型フリップフロップ342、341より入力された16ビットのデータのうち、LSB側の15ビットが、それぞれ対応するスイッチ351-0乃至351-14の接点aに供給されている。各スイッチの接点bには、1ビットだけMSB側のデータが供給されている。デコーダ352は、コントローラ346より供給される信号delete positionにより指定されるビットを削除して、15ビットのデータとして出力するようになされている。

【0364】図26は、LSBから第13番目のビットがディリートされる状態を示している。従って、この場合、スイッチ351-0乃至スイッチ351-11が接点a側に切り替えられ、LSBから第12番目までの12ビットが、そのまま選択、出力されている。また、スイッチ351-12乃至351-14は、それぞれ接点b側に切り替えられているので、第14番目乃至第16番目のデータが、第13番目乃至第15番目のビットのデータとして選択、出力される。

【0365】図23のスタッフ回路323および図26のディリート回路343の入力が16ビットとなっているのは、それぞれ図22のコンバータ212のスタッフ回路323の入力が、D型フリップフロップ322、321より供給される16ビットとされており、また、図25のコンバータ202においても、ディリート回路343の入力が、D型フリップフロップ342、341により16ビットとされているためである。図22において、スタッフ回路323の出力する17ビットをバレルシフタ324でバレルシフトすることにより、例えば8ビットを最終的に選択、出力しているのと同様に、図25のコンバータ202においても、ディリート回路343の出力する15ビットのデータを、バレルシフタ344で所定量だけバレルシフトすることにより、8ビットのデータとしている。

【0366】図27は、コンバータ212の他の構成例を表している。この構成例においては、カウンタ361が入力データのうち、連続する0のビットの数をカウントし、そのカウント結果をコントローラ326に出力するようになされている。コントローラ326は、例えばカウンタ361が連続する0のビットを22個カウント

したとき、信号stuff positionをスタップ回路323に出力する。また、このとき、コントローラ326は、カウンタ361をリセットし、再び連続する0のビットの数をカウンタ361にカウントさせる。

【0367】その他の構成と動作は、図22における場合と同様である。

【0368】図28は、コンバータ202の他の構成例を表している。この構成例においては、入力データのうち、連続する0の数をカウンタ371がカウントし、そのカウント結果をコントローラ346に出力するようになされている。カウンタ371のカウント値が22に達したとき、コントローラ346は、信号delete positionをディリート回路343に出力するとともに、カウンタ371をリセットし、再び新たな連続する0のビットの数をカウンタ371にカウントさせる。その他の構成は、図25における場合と同様である。

【0369】このように、この構成例においては、所定のパターン（データ“0”の連続する数）に基づいて、マーカービットとしてのデータ“1”が挿入され、また、削除されることになる。

【0370】図27と図28に示す構成は、図22と図25に示す構成よりも効率的な処理が可能となる。但し、変換後の長さが元の履歴情報に依存することになる。

【0371】図29は、ユーザデータフォーマット213の構成例を表している。この例においては、コントローラ383がコンバータ212とユーザデータフォーマット213との間に配置されているバッファメモリ（図示せず）に読み出しアドレスを出力すると、そこから読み出されたデータが、ユーザデータフォーマット213のスイッチ382の接点a側に供給される。ROM381には、ユーザデータスタートコード、データIDなどのuser_data0を生成するのに必要なデータが記憶されている。コントローラ313は、所定のタイミングにおいて、スイッチ382を接点a側または接点b側に切り替え、ROM381に記憶されているデータ、またはコンバータ212より供給されるデータを適宜選択し、出力する。これにより、user_data0のフォーマットのデータが符号化装置106に出力される。

【0372】なお、図示は省略するが、ユーザデータデコード201は、図29のROM381より読み出され、挿入されたデータを削除するスイッチを介して、入力データを出力するようにすることで実現することができる。

【0373】図30は、例えば映像編集スタジオにおいて、複数のトランスコード101-1乃至101-Nが直列に接続されて使用される状態を示している。各トランスコード101-i（i=1乃至N）の履歴情報多重化装置103-iは、上述した符号化パラメータ用の領域の最も古い符号化パラメータが記録されている区

画に、自己が用いた最新の符号化パラメータを上書きする。このことにより、ベースバンドの画像データには、同一のマクロブロックに対応する直近の4世代分の符号化パラメータ（世代履歴情報）が記録されることになる（図18）。

【0374】各符号化装置106-iのエンコーダ121-i（図19）は、その可変長符号化回路58において、履歴情報分離装置105-iから供給される今回用いる符号化パラメータに基づいて、量子化回路57より供給されるビデオデータを符号化する。このようにして生成されるビットストリーム（例えば、picture_header()）中に、その現符号化パラメータは多重化される。

【0375】可変長符号化回路58はまた、履歴エンコーディング装置107-iより供給されるユーザデータ（世代履歴情報を含む）を、出力するビットストリーム中に多重化する（図18に示すような埋め込み処理ではなく、ビットストリーム中に多重化する）。そして、符号化装置106-iの出力するビットストリームは、SDTI(Serial Data Transfer Interface)108-iを介して、後段のトランスコード101-(i+1)に入力される。

【0376】トランスコード101-iとトランスコード101-(i+1)は、それぞれ図15に示すように構成されている。従って、その処理は、図15を参照して説明した場合と同様となる。

【0377】実際の符号化パラメータの履歴を利用した符号化として、現在Iピクチャとして符号化されていたものを、PもしくはBピクチャに変更したい場合、過去の符号化パラメータの履歴を見て、過去にPもしくはBピクチャであった場合を探し、これらの履歴が存在した場合は、その動きベクトルなどのパラメータを利用して、ピクチャタイプを変更する。反対に過去に履歴がない場合は、動き検出を行わないピクチャタイプの変更を断念する。もちろん履歴がない場合であっても、動き検出を行えばピクチャタイプを変更できる。

【0378】図18に示すフォーマットの場合、4世代分の符号化パラメータを埋め込むようにしたが、I、P、Bの各ピクチャタイプのパラメータを埋め込むようにすることもできる。図31は、この場合のフォーマットの例を示している。この例では、同一のマクロブロックが、過去にピクチャタイプの変更を伴って符号化されたときにおける、ピクチャタイプ毎に1世代分の符号化パラメータ（ピクチャ履歴情報）が記録される。したがって、図16に示したデコーダ111、および図19に示したエンコーダ121は、現在（最新）、第3世代、第2世代、および第1世代の符号化パラメータの代わりに、Iピクチャ、Pピクチャ、およびBピクチャに対応する1世代分の符号化パラメータを入出力することになる。

【0379】また、この例の場合、Cb[1][x]とCr[1][x]の空き領域は利用しないので、Cb[1][x]とCr[1][x]の領域を有さない4:2:0フォーマットの画像データにも本発明を適用することができる。

【0380】この例の場合、復号装置102は、符号化パラメータを復号と同時に取り出し、ピクチャタイプを判定して、画像信号のピクチャタイプに対応した場所に符号化パラメータを書き込んで(多重化して)履歴情報分離装置105に出力する。履歴情報分離装置105は、符号化パラメータを分離し、これから符号化

したいピクチャタイプと、入力された過去の符号化パラメータを考慮して、ピクチャタイプを変更しながら再符号化を行うことができる。

【0381】次に、各トランスコード101において、変更が可能なピクチャタイプを判定する処理について、図32のフローチャートを参照して説明する。なお、トランスコード101におけるピクチャタイプの変更は、過去の動きベクトルを利用するので、この処理は動き検出を行わないで実行されることを前提としている。また、以下に説明する処理は、履歴情報分離装置10

5により実行される。

【0382】ステップS1において、ピクチャタイプ毎に1世代分の符号化パラメータ(ピクチャ履歴情報)が履歴情報分離装置105に入力される。

【0383】ステップS2において、履歴情報分離装置105は、ピクチャ履歴情報の中に、Bピクチャに変更したときの符号化パラメータが存在するか否かを判定する。ピクチャ履歴情報にBピクチャに変更したときの符号化パラメータが存在すると判定された場合、ステップS3に進む。

【0384】ステップS3において、履歴情報分離装置105は、ピクチャ履歴情報の中に、Pピクチャに変更したときの符号化パラメータが存在するか否かを判定する。ピクチャ履歴情報にPピクチャに変更したときの符号化パラメータが存在すると判定された場合、ステップS4に進む。

【0385】ステップS4において、履歴情報分離装置105は、変更可能なピクチャタイプがIピクチャ、Pピクチャ、およびBピクチャであると判断する。

【0386】ステップS3において、ピクチャ履歴情報にPピクチャに変更したときの符号化パラメータが存在しないと判定された場合、ステップS5に進む。

【0387】ステップS5において、履歴情報分離装置105は、変更可能なピクチャタイプがIピクチャ、およびBピクチャであると判断する。さらに、履歴情報分離装置105は、特殊処理(Bピクチャの履歴情報に含まれる後方予測ベクトルを使わず、前方予測ベクトルだけを使う)を施すことにより、擬似的にPピクチャに変更可能であると判断する。

【0388】ステップS2において、ピクチャ履歴情報

にBピクチャに変更したときの符号化パラメータが存在しないと判定された場合、ステップS6に進む。

【0389】ステップS6において、履歴情報分離装置105は、ピクチャ履歴情報にPピクチャに変更したときの符号化パラメータが存在するか否かを判定する。ピクチャ履歴情報にPピクチャに変更したときの符号化パラメータが存在すると判定された場合、ステップS7に進む。

【0390】ステップS7において、履歴情報分離装置105は、変更可能なピクチャタイプがIピクチャ、およびPピクチャであると判断する。さらに、履歴情報分離装置105は、特殊処理(Pピクチャに履歴情報に含まれる前方予測ベクトルだけを使う)を施すことにより、Bピクチャに変更可能であると判断する。

【0391】ステップS6において、ピクチャ履歴情報にPピクチャに変更したときの符号化パラメータが存在しないと判定された場合、ステップS8に進む。ステップS8において、履歴情報分離装置105は、動きベクトルが存在しないので、変更可能なピクチャタイプがIピクチャだけである(IピクチャなのでIピクチャ以外には変更できない)と判断する。

【0392】ステップS4、S5、S7、S8の処理の次にステップS9において、履歴情報分離装置105は、変更可能なピクチャタイプを表示装置(図示せず)に表示してユーザに通知する。

【0393】図33は、ピクチャタイプ変更の例を示している。ピクチャタイプを変更する場合、GOPを構成するフレーム数を変更される。すなわち、この例の場合、N=15(GOPのフレーム数N=15)、M=3(GOP内のI、またはPピクチャの出現周期M=3)のフレームから構成される4MbpsのLong GOP(第1世代)から、N=1、M=1のフレームで構成される50MbpsのShort GOP(第2世代)に変換され、再度、N=15、M=3のフレームから構成される4MbpsのLong GOP(第3世代)に変換されている。なお、図中において破線は、GOPの境界を示している。

【0394】第1世代から第2世代にピクチャタイプが変更される場合において、上述した変更可能ピクチャタイプ判定処理の説明から明らかなように、全てのフレームは、ピクチャタイプをIピクチャに変更することが可能である。このピクチャタイプ変更のとき、動画像(第0世代)が第1世代に変換されたときに演算された全ての動きベクトルは、ピクチャ履歴情報に保存された(残された)状態となる。次に、再度Long GOPに変換される(第2世代から第3世代にピクチャタイプが変更される)場合、第0世代から第1世代に変換されたときのピクチャタイプ毎の動きベクトルが保存されているので、これを再利用することにより、画質劣化を抑えて、再度、Long GOPに変換することが可能となる。

【0395】図34は、ピクチャタイプ変更の他の例を示している。この例の場合、N=14、M=2である4MbpsのL

ong GOP (第1世代) から、 $N=2$, $M=2$ である18MbpsのShort GOP (第2世代) に変換され、さらに、 $N=1$, $M=1$ であるフレーム数が1の50MbpsのShort GOP (第3世代) に変換されて、1Mbpsの、フレーム数 N がランダムなGOP (第4世代) に変換される。

【0396】この例においても、第0世代から第1世代に変換されたときのピクチャタイプ毎の動きベクトルが、第3世代から第4世代への変換のときまで保存される。そこで、図34に示すように、複雑にピクチャタイプを変更しても、保存されている符号化パラメータを再利用されることにより、画質劣化を小さく抑えることができる。さらに、保存されている符号化パラメータの量子化スケールを有効に利用すれば画質劣化の少ない符号化を実現できる。

【0397】この量子化スケールの再利用について、図35を参照して説明する。図35は、所定のフレームが、第1世代から第4世代まで常に、Iピクチャに変換されており、ビットレートだけが、4Mbps、18Mbps、または50Mbpsに変更されていることを示している。

【0398】例えば、第1世代(4Mbps)から第2世代(18Mbps)への変換の際に、ビットレートの高速化に伴って、細かい量子化スケールで再符号化しても画質は向上しない。なぜならば、過去において粗い量子化ステップで量子化されたデータは、復元しないからである。したがって、図35に示すように、途中でビットレートが高速化しても、それに伴って細かい量子化ステップで量子化することは、情報量が増加するだけであって画質の向上には繋がらない。したがって、過去のもっとも粗い(大きい)量子化スケールを維持するように制御すれば、最も無駄が無く、効率的な符号化が可能となる。

【0399】なお、第3世代から第4世代への変換時には、ビットレートは、50Mbpsから4Mbpsに低下されているが、この場合にも、過去のもっとも粗い(大きい)量子化スケールが維持される。

【0400】上述したように、ビットレートが変更されるときは、過去の量子化スケールの履歴を利用して符号化することは非常に有効である。

【0401】この量子化制御処理について、図36のフローチャートを参照して説明する。ステップS11において、ヒストリ情報分離装置105は、入力されたピクチャ履歴情報に、いまから変換するピクチャタイプの符号化パラメータが存在するか否かを判定する。変換するピクチャタイプの符号化パラメータが存在すると判定された場合、ステップS12に進む。

【0402】ステップS12において、ヒストリ情報分離装置105は、ピクチャ履歴情報の対象となる符号化パラメータから、history_q_scale_codeを抽出する。

【0403】ステップS13において、ヒストリ情報分離装置105は、送信バッファ59から量子化回路57にフィードバックされるバッファ残量に基づいて、feed

back_q_scale_codeを演算する。

【0404】ステップS14において、ヒストリ情報分離装置105は、history_q_scale_codeがfeedback_q_scale_codeよりも大きい(粗い)か否かを判定する。history_q_scale_codeがfeedback_q_scale_codeよりも大きいと判定された場合、ステップS15に進む。

【0405】ステップS15において、ヒストリ情報分離装置105は、量子化スケールとしてhistory_q_scale_codeを量子化回路57に出力する。量子化回路57は、history_q_scale_codeを用いて量子化を実行する。

【0406】ステップS16において、フレームに含まれる全てのマクロブロックが量子化されたか否かが判定される。全てのマクロブロックがまだ量子化されていないと判定された場合、ステップS12に戻り、ステップS12乃至S16の処理が、全てのマクロブロックが量子化されるまで繰り返される。

【0407】ステップS14において、history_q_scale_codeがfeedback_q_scale_codeよりも大きくない(細かい)いと判定された場合、ステップS17に進む。

【0408】ステップS17において、ヒストリ情報分離装置105は、量子化スケールとしてfeedback_q_scale_codeを量子化回路57に出力する。量子化回路57は、feedback_q_scale_codeを用いて量子化を実行する。

【0409】ステップS11において、変換するピクチャタイプの符号化パラメータが、ヒストリ情報中に存在しないと判定された場合、ステップS18に進む。

【0410】ステップS18において、ヒストリ情報分離装置105は、送信バッファ59から量子化回路57にフィードバックされるバッファ残量に基づいて、feedback_q_scale_codeを演算する。

【0411】ステップS19において、量子化回路57は、Feedback_q_scale_codeを用いて量子化を実行する。

【0412】ステップS20において、フレームに含まれる全てのマクロブロックが量子化されたか否かが判定される。全てのマクロブロックがまだ量子化されていないと判定された場合、ステップS18に戻り、ステップS18乃至S20の処理が、全てのマクロブロックが量子化されるまで繰り返される。

【0413】なお、本実施の形態におけるトランスコーダ101の内部においては、上述したように、復号側と符号側が粗結合されており、符号化パラメータを画像データに多重化させて伝送させたが、図37に示すように、復号装置102と符号化装置106を直接接続する(密結合する)ようにしてもよい。

【0414】図15において説明したトランスコーダ101は、第1世代から第3世代の過去の符号化パラメータを符号化装置106に供給するために、ベースバンド

ビデオデータに過去の符号化パラメータを多重化して伝送するようにしていた。しかしながら、本発明においては、ベースバンドビデオデータに過去の符号化パラメータを多重化する技術は必須ではなく、図 37 に示されたように、ベースバンドビデオデータとは異なる伝送路（たとえばデータ転送バス）を使用して、過去の符号化パラメータを伝送するようにしても良い。

【0415】つまり、図 37 に示した、復号装置 102、ヒストリデコーディング装置 104、符号化装置 106 及びヒストリエンコーディング装置 107 は、図 105 において説明した復号装置 102、ヒストリデコーディング装置 104、符号化装置 106 及びヒストリエンコーディング装置 107 とまったく同じ機能及び構成を有している。

【0416】復号装置 102 の可変長復号回路 112 は、第 3 世代の符号化ストリーム ST(3rd) のシーケンス層、GOP 層、ピクチャ層、スライス層及びマクロブロック層から、第 3 世代の符号化パラメータを抽出し、それを、ヒストリエンコーディング装置 107 及び符号化装置 106 のコントローラ 70 にそれぞれ供給する。ヒストリエンコーディング装置 107 は、受け取った第 3 世代の符号化パラメータをピクチャ層のユーザデータエリアに記述できるように converted_history_stream() に変換し、converted_history_stream() をユーザデータとして符号化装置 106 の可変長符号化回路 58 に供給する。

【0417】さらに可変長復号回路 112 は、第 3 世代の符号化ストリームのピクチャ層のユーザデータエリアから、第 1 世代の符号化パラメータ及び第 2 の符号化パラメータを含んでいるユーザデータ user_data を抽出し、ヒストリデコーディング装置 104 及び符号化装置 106 の可変長符号化回路 58 に供給する。ヒストリデコーディング装置 104 は、ユーザデータエリアに converted_history_stream() として記述されたヒストリストリームから、第 1 世代の符号化パラメータ及び第 2 世代の符号化パラメータを抽出し、それを符号化装置 106 のコントローラに供給する。

【0418】符号化装置 106 のコントローラ 70 は、ヒストリデコーディング装置 104 から受け取った第 1 世代及び第 2 世代の符号化パラメータと、符号化装置 102 から受け取った第 3 世代の符号化パラメータとに基づいて、符号化装置 106 の符号化処理をコントロールする。

【0419】符号化装置 106 の可変長符号化回路 58 は、復号装置 102 から第 1 世代の符号化パラメータ及び第 2 の符号化パラメータを含んでいるユーザデータ user_data を受け取るとともに、ヒストリエンコーディング装置 107 から第 3 世代の符号化パラメータを含んでいるユーザデータ user_data を受け取り、それらのユーザデータをヒストリ情報として、第 4 世代の符号化スト

リームのピクチャ層のユーザデータエリアに記述する。

【0420】図 38 は、MPEG のビデオストリームをデコードするためのシンタックスを表わした図である。デコードは、このシンタックスに従って MPEG ビットストリームをデコードすることによって、ビットストリームから意味のある複数のデータ項目（データエレメント）を抽出する。以下に説明するシンタックスは、図において、その関数や条件文は細活字で表わされ、そのデータエレメントは、太活字で表されている。データ項目は、その名称、ビット長、及びそのタイプと伝送順序を示すニーモニック (Mnemonic) で記述されている。

【0421】まず、この図 38 に示されているシンタックスにおいて使用されている関数について説明する。

【0422】next_start_code() 関数は、ビットストリーム中に記述されているスタートコードを探すための関数である。この図 38 に示されたシンタックスにおいて、この next_start_code() 関数の次に、sequence_header() 関数と sequence_extension() 関数とが順に配置されているので、このビットストリームには、この sequence_header() 関数と sequence_extension() 関数によって定義されたデータエレメントが記述されている。従って、ビットストリームのデコード時には、この next_start_code() 関数によって、sequence_header() 関数と sequence_extension() 関数の先頭に記述されているスタートコード（データエレメントの一種）をビットストリーム中から見つけ、それを基準にして、sequence_header() 関数と sequence_extension() 関数をさらに見つけ、それらによって定義された各データエレメントをデコードする。

【0423】尚、sequence_header() 関数は、MPEG ビットストリームのシーケンス層のヘッダデータを定義するための関数であって、sequence_extension() 関数は、MPEG ビットストリームのシーケンス層の拡張データを定義するための関数である。

【0424】sequence_extension() 関数の次に配置されている do { } while 構文は、while 文によって定義されている条件が真である間、do 文の { } 内の関数に基いて記述されたデータエレメントをデータストリーム中から抽出するための構文である。すなわち、do { } while 構文によって、while 文によって定義されている条件が真である間、ビットストリーム中から、do 文内の関数に基いて記述されたデータエレメントを抽出するデコード処理が行われる。

【0425】この while 文に使用されている nextbits() 関数は、ビットストリーム中に現れるビット又はビット列と、次にデコードされるデータエレメントとを比較するための関数である。この図 38 のシンタックスの例では、nextbits() 関数は、ビットストリーム中のビット列とビデオシーケンスの終わりを示す sequence_end_code とを比較し、ビットストリーム中のビット列と sequence_end_code とが一致しないときに、この while 文の条件が

真となる。従って、sequence_extension()関数の次に配置されている do{ }while構文は、ビットストリーム中に、ビデオシーケンスの終わりを示すsequence_end_codeが現れない間、do文中の関数によって定義されたデータエレメントがビットストリーム中に記述されていることを示している。

【0426】ビットストリーム中には、sequence_extension()関数によって定義された各データエレメントの次には、extension_and_user_data(0)関数によって定義されたデータエレメントが記述されている。このextension_and_user_data(0)関数は、MPEGビットストリームのシーケンス層の拡張データとユーザデータを定義するための関数である。

【0427】このextension_and_user_data(0)関数の次に配置されている do{ }while構文は、while文によって定義されている条件が真である間、do文の{ }内の関数に基づいて記述されたデータエレメントを、ビットストリーム中から抽出するための関数である。このwhile文において使用されているnextbits()関数は、ビットストリーム中に現れるビット又はビット列と、picture_start_code又はgroup_start_codeとの一致を判断するための関数であって、ビットストリーム中に現れるビット又はビット列と、picture_start_code又はgroup_start_codeとが一致する場合には、while文によって定義された条件が真となる。よって、このdo{ }while構文は、ビットストリーム中において、picture_start_code又はgroup_start_codeが現れた場合には、そのスタートコードの次に、do文中の関数によって定義されたデータエレメントのコードが記述されているので、このpicture_start_code又はgroup_start_codeによって示されるスタートコードを探し出すことによって、ビットストリーム中からdo文中に定義されたデータエレメントを抽出することができる。

【0428】このdo文の最初に記述されているif文は、ビットストリーム中にgroup_start_codeが現れた場合、という条件を示している。このif文による条件が真である場合には、ビットストリーム中には、このgroup_start_codeの次にgroup_of_picture_header(1)関数及びextension_and_user_data(1)関数によって定義されているデータエレメントが順に記述されている。

【0429】このgroup_of_picture_header(1)関数は、MPEGビットストリームのGOP層のヘッダデータを定義するための関数であって、extension_and_user_data(1)関数は、MPEGビットストリームのGOP層の拡張データ(extension_data)及びユーザデータ(user_data)を定義するための関数である。

【0430】さらに、このビットストリーム中には、group_of_picture_header(1)関数及びextension_and_user_data(1)関数によって定義されているデータエレメントの次に、picture_header()関数とpicture_coding_exten

sion()関数によって定義されたデータエレメントが記述されている。もちろん、先に説明したif文の条件が真とならない場合には、group_of_picture_header(1)関数及びextension_and_user_data(1)関数によって定義されているデータエレメントは記述されていないので、extension_and_user_data(0)関数によって定義されているデータエレメントの次に、picture_header()関数とpicture_coding_extension()関数によって定義されたデータエレメントが記述されている。

10 【0431】このpicture_header()関数は、MPEGビットストリームのピクチャ層のヘッダデータを定義するための関数であって、picture_coding_extension()関数は、MPEGビットストリームのピクチャ層の第1の拡張データを定義するための関数である。

【0432】次のwhile文は、このwhile文によって定義されている条件が真である間、次のif文の条件判断を行うための関数である。このwhile文において使用されているnextbits()関数は、ビットストリーム中に現れるビット列と、extension_start_code又はuser_data_start_codeとの一致を判断するための関数であって、ビットストリーム中に現れるビット列と、extension_start_code又はuser_data_start_codeとが一致する場合には、このwhile文によって定義された条件が真となる。

20 【0433】第1のif文は、ビットストリーム中に現れるビット列とextension_start_codeとの一致を判断するための関数である。ビットストリーム中に現れるビット列と32ビットのextension_start_codeとが一致する場合には、ビットストリーム中において、extension_start_codeの次にextension_data(2)関数によって定義されるデータエレメントが記述されている。

30 【0434】第2のif文は、ビットストリーム中に現れるビット列とuser_data_start_codeとの一致を判断するための構文であって、ビットストリーム中に現れるビット列と32ビットのuser_data_start_codeとが一致する場合には、第3のif文の条件判断が行われる。このuser_data_start_codeは、MPEGビットストリームのピクチャ層のユーザデータエリアの開始を示すためのスタートコードである。

40 【0435】第3のif文は、ビットストリーム中に現れるビット列とHistory_Data_IDとの一致を判断するための構文である。ビットストリーム中に現れるビット列とこの32ビットのHistory_Data_IDとが一致する場合には、このMPEGビットストリームのピクチャ層のユーザデータエリアにおいて、この32ビットのHistory_Data_IDによって示されるコードの次に、converted_history_stream()関数によって定義されるデータエレメントが記述されている。

50 【0436】converted_history_stream()関数は、MPEG符号化時に使用したあらゆる符号化パラメータを伝送するための履歴情報及び履歴データを記述するための関数

である。このconverted_history_stream()関数によって定義されているデータ要素の詳細は、図40乃至図47を参照して、history_stream()として後述する。また、このHistory_Data_IDは、MPEGビットストリームのピクチャ層のユーザデータエリアに記述されたこの履歴情報及び履歴データが記述されている先頭を示すためのスタートコードである。

【0437】else文は、第3のif文において、条件が非真であることを示すための構文である。従って、このMPEGビットストリームのピクチャ層のユーザデータエリアにおいて、converted_history_stream()関数によって定義されたデータ要素が記述されていない場合には、user_data()関数によって定義されたデータ要素が記述されている。

【0438】図38において、履歴情報は、converted_history_stream()に記述され、user_data()に記述される訳ではないが、このconverted_history_stream()は、MPEG規格のuser_dataの一種として記述される。そこで、本明細書中においては、場合によって、履歴情報がuser_dataに記述されるとも説明するが、それは、MPEG規格のuser_dataの一種として記述されるということを意味する。

【0439】picture_data()関数は、MPEGビットストリームのピクチャ層のユーザデータの次に、スライス層及びマクロブロック層に関するデータ要素を記述するための関数である。通常は、このpicture_data()関数によって示されるデータ要素は、ビットストリームのピクチャ層のユーザデータエリアに記述されたconverted_history_stream()関数によって定義されるデータ要素又はuser_data()関数によって定義されたデータ要素の次に記述されているが、ピクチャ層のデータ要素を示すビットストリーム中に、extension_start_code又はuser_data_start_codeが存在しない場合には、このpicture_data()関数によって示されるデータ要素は、picture_coding_extension()関数によって定義されるデータ要素の次に記述されている。

【0440】このpicture_data()関数によって示されるデータ要素の次には、sequence_header()関数とsequence_extension()関数とによって定義されたデータ要素が順に配置されている。このsequence_header()関数とsequence_extension()関数によって記述されたデータ要素は、ビデオストリームのシーケンスの先頭に記述されたsequence_header()関数とsequence_extension()関数によって記述されたデータ要素と全く同じである。このように同じデータをストリーム中に記述する理由は、ビットストリーム受信装置側でデータストリームの途中(例えばピクチャ層に対応するビットストリーム部分)から受信が開始された場合に、シーケンス層のデータを受信できなくなり、ストリームを

デコード出来なくなることを防止するためである。

【0441】この最後のsequence_header()関数とsequence_extension()関数とによって定義されたデータ要素の次、つまり、データストリームの最後には、シーケンスの終わりを示す32ビットのsequence_end_codeが記述されている。

【0442】以上のシンタックスの基本的な構成の概略を示すと、図39に示すようになる。

【0443】次に、converted_history_stream()関数によって定義されたヒストリストリームに関して説明する。

【0444】このconverted_history_stream()は、MPEGのピクチャ層のユーザデータエリアに履歴情報を示すヒストリストリームを挿入するための関数である。尚、

「converted」の意味は、スタートエミュレーションを防止するために、ユーザエリアに挿入すべき履歴データから構成される履歴ストリームの少なくとも22ビット毎にマーカービット(1ビット)を挿入する変換処理を行ったストリームであることを意味している。

【0445】このconverted_history_stream()は、以下に説明する固定長の履歴ストリーム(図40乃至図46)又は可変長の履歴ストリーム(図47)のいずれかの形式で記述される。エンコーダ側において固定長の履歴ストリームを選択した場合には、デコーダ側において履歴ストリームから各データ要素をデコードするための回路及びソフトウェアが簡単になるというメリットがある。一方、エンコーダ側において可変長の履歴ストリームを選択した場合には、エンコーダにおいてピクチャ層のユーザエリアに記述される履歴情報(データ要素)を必要に応じて任意に選択することができるので、履歴ストリームのデータ量を少なくすることができ、その結果、符号化されたビットストリーム全体のデータレートを低減することができる。

【0446】本発明において説明する「履歴ストリーム」、「ヒストリストリーム」、「履歴情報」、「ヒストリ情報」、「履歴データ」、「ヒストリデータ」、「履歴パラメータ」、「ヒストリパラメータ」とは、過去の符号化処理において使用した符号化パラメータ(又はデータ要素)を意味し、現在の(最終段の)符号化処理において使用した符号化パラメータを意味するものではない。例えば、第1世代の符号化処理において、あるピクチャをIピクチャで符号化して伝送し、次なる第2世代の符号化処理において、このピクチャを今度はPピクチャとして符号化して伝送し、さらに、第3世代の符号化処理において、このピクチャをBピクチャで符号化して伝送する例をあげて説明する。

【0447】第3世代の符号化処理において使用した符号化パラメータが、第3世代の符号化処理において生成された符号化ビットストリームのシーケンス層、GOP層、ピクチャ層、スライス層及びマクロブロック層の所

定位置に記述されている。一方、過去の符号化処理である第1世代及び第2世代の符号化処理において使用した符号化パラメータは、第3世代の符号化処理において使用した符号化パラメータが記述されるシーケンス層やGOP層に記述されるのではなく、既に説明したシンタックスに従って、符号化パラメータの履歴情報として、ピクチャ層のユーザデータエリアに記述される。

【0448】まず、固定長の履歴ストリームシンタックスについて図40乃至図46を参照して説明する。

【0449】最終段（例えば第3世代）の符号化処理において生成されたビットストリームのピクチャ層のユーザデータエリアには、まず最初に、過去（例えば第1世代及び第2世代）の符号化処理において使用されていたシーケンス層のシーケンスヘッダに含められる符号化パラメータが、履歴ストリームとして挿入される。尚、過去の符号化処理において生成されたビットストリームのシーケンス層のシーケンスヘッダ等の履歴情報は、最終段の符号化処理において生成されたビットストリームのシーケンス層のシーケンスヘッダに挿入されることは無いという点に注意すべきである。

【0450】過去の符号化処理で使用したシーケンスヘッダ(sequence_header)に含められるデータエレメントは、sequence_header_code、sequence_header_present_flag、horizontal_size_value、marker_bit、vertical_size_value、aspect_ratio_information、frame_rate_code、bit_rate_value、VBV_buffer_size_value、constrained_parameter_flag、load_intra_quantiser_matrix、load_non_intra_quantiser_matrix、intra_quantiser_matrix、及びnon_intra_quantiser_matrix等から構成される。

【0451】sequence_header_codeは、シーケンス層のスタート同期コードを表すデータである。sequence_header_present_flagは、sequence_header内のデータが有効か無効かを示すデータである。horizontal_size_valueは、画像の水平方向の画素数の下位12ビットから成るデータである。marker_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。vertical_size_valueは、画像の縦のライン数の下位12ビットからなるデータである。aspect_ratio_informationは、画素のアスペクト比（縦横比）または表示画面アスペクト比を表すデータである。frame_rate_codeは、画像の表示周期を表すデータである。

【0452】bit_rate_valueは、発生ビット量に対する制限のためのビット・レートの下位18ビット(400bsp単位で切り上げる)データである。VBV_buffer_size_valueは、発生符号量制御用の仮想バッファ（ビデオバッファペリファイヤー）の大きさを決める値の下位10ビットデータである。constrained_parameter_flagは、各パラメータが制限以内であることを示すデータである。load_intra_quantiser_matrixは、イントラMB用量子化マトリ

ックス・データの存在を示すデータである。load_non_intra_quantiser_matrixは、非イントラMB用量子化マトリックス・データの存在を示すデータである。intra_quantiser_matrixは、イントラMB用量子化マトリックスの値を示すデータである。non_intra_quantiser_matrixは、非イントラMB用量子化マトリックスの値を表すデータである。

【0453】最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザデータエリアには、過去の符号化処理において使用されたシーケンス層のシーケンスエクステンションを表わすデータエレメントが、履歴ストリームとして記述される。

【0454】この過去の符号化処理で使用したシーケンスエクステンション(sequence_extension)を表わすデータエレメントは、extension_start_code、extension_start_code_identifier、sequence_extension_present_flag、profile_and_level_indication、progressive_sequence、chroma_format、horizontal_size_extension、vertical_size_extension、bit_rate_extension、vbv_buffer_size_extension、low_delay、frame_rate_extension_n、及びframe_rate_extension_d等のデータエレメントである。

【0455】extension_start_codeは、エクステンションデータのスタート同期コードを表すデータである。extension_start_code_identifierは、どの拡張データが送られるかを示すデータである。sequence_extension_present_flagは、シーケンスエクステンション内のデータが有効であるか無効であることを示すデータである。profile_and_level_indicationは、ビデオデータのプロファイルとレベルを指定するためのデータである。progressive_sequenceは、ビデオデータが順次走査であることを示すデータである。chroma_formatは、ビデオデータの色差フォーマットを指定するためのデータである。

【0456】horizontal_size_extensionは、シーケンスヘッダのhorizontal_size_valueに加える上位2ビットのデータである。vertical_size_extensionは、シーケンスヘッダのvertical_size_valueに加える上位2ビットのデータである。bit_rate_extensionは、シーケンスヘッダのbit_rate_valueに加える上位12ビットのデータである。vbv_buffer_size_extensionは、シーケンスヘッダのvbv_buffer_size_valueに加える上位8ビットのデータである。low_delayは、Bピクチャを含まないことを示すデータである。frame_rate_extension_nは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを得るためのデータである。frame_rate_extension_dは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを得るためのデータである。

【0457】続いて、ビットストリームのピクチャ層のユーザデータエリアには、過去の符号化処理において使用され

たシーケンス層のシーケンスディスプレイエクステンションを表わすデータエレメントが、履歴ストリームとして記述される。

【0458】このシーケンスディスプレイエクステンション(sequence_display_extension)として記述されているデータエレメントは、extension_start_code、extension_start_code_identifier、sequence_display_extension_present_flag、video_format、colour_description、colour_primaries、transfer_characteristics、matrix_coefficients、display_horizontal_size、及びdisplay_vertical_sizeから構成される。

【0459】extension_start_codeは、エクステンションデータのスタート同期コードを表すデータである。extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。sequence_display_extension_present_flagは、シーケンスディスプレイエクステンション内のデータエレメントが有効か無効かを示すデータである。video_formatは、原信号の映像フォーマットを表すデータである。color_descriptionは、色空間の詳細データがあることを示すデータである。color_primariesは、原信号の色特性の詳細を示すデータである。transfer_characteristicsは、光電変換がどのように行われたのかの詳細を示すデータである。matrix_coefficientsは、原信号が光の三原色からどのように変換されたかの詳細を示すデータである。display_horizontal_sizeは、意図するディスプレイの活性領域(水平サイズ)を表すデータである。display_vertical_sizeは、意図するディスプレイの活性領域(垂直サイズ)を表すデータである。

【0460】続いて、最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において生成されたマクロブロックの位相情報を示すマクロブロックアサインメントデータ(macroblock_assignment_in_user_data)が、履歴ストリームとして記述される。

【0461】このマクロブロックの位相情報を示すmacroblock_assignment_in_user_dataは、macroblock_assignment_present_flag、v_phase、h_phase等のデータエレメントから構成される。

【0462】このmacroblock_assignment_present_flagは、macroblock_assignment_in_user_data内のデータエレメントが有効か無効かを示すデータである。v_phaseは、画像データからマクロブロックを切り出す際の垂直方向の位相情報を示すデータである。h_phaseは、画像データからマクロブロックを切り出す際の水平方向の位相情報を示すデータである。

【0463】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたGOP層のGOPヘッダを表わすデータエレメントが、履歴ストリームとし

て記述されている。

【0464】このGOPヘッダ(group_of_picture_header)を表わすデータエレメントは、group_start_code、group_of_picture_header_present_flag、time_code、closed_gop、及びbroken_linkから構成される。

【0465】group_start_codeは、GOP層の開始同期コードを示すデータである。group_of_picture_header_present_flagは、group_of_picture_header内のデータエレメントが有効であるか無効であることを示すデータである。time_codeは、GOPの先頭ピクチャのシーケンスの先頭からの時間を示すタイムコードである。closed_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

【0466】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャ層のピクチャヘッダを表わすデータエレメントが、履歴ストリームとして記述されている。

【0467】このピクチャヘッダ(picture_header)に関するデータエレメントは、picture_start_code、temporal_reference、picture_coding_type、v_bv_delay、full_pel_forward_vector、forward_f_code、full_pel_backward_vector、及びbackward_f_codeから構成される。

【0468】具体的には、picture_start_codeは、ピクチャ層の開始同期コードを表すデータである。temporal_referenceは、ピクチャの表示順を示す番号でGOPの先頭でリセットされるデータである。picture_coding_typeは、ピクチャタイプを示すデータである。v_bv_delayは、ランダムアクセス時の仮想バッファの初期状態を示すデータである。full_pel_forward_vectorは、順方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。forward_f_codeは、順方向動きベクトル探索範囲を表すデータである。full_pel_backward_vectorは、逆方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。backward_f_codeは、逆方向動きベクトル探索範囲を表すデータである。

【0469】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャ層のピクチャコーディングエクステンションが、履歴ストリームとして記述されている。

【0470】このピクチャコーディングエクステンション(picture_coding_extension)に関するデータエレメントは、extension_start_code、extension_start_code_identifier、f_code[0][0]、f_code[0][1]、f_code[1][0]、f_code[1][1]、intra_dc_precision、picture_structure、top_field_first、frame_predictive_frame_dc

t, concealment_motion_vectors, q_scale_type, intra_vlc_format, alternate_scan, repeat_firt_field, chroma_420_type, progressive_frame, composite_display_flag, v_axis, field_sequence, sub_carrier, burst_amplitude, 及び sub_carrier_phase から構成される。

【0471】 extension_start_code は、ピクチャ層のエクステンションデータのスタートを示す開始コードである。extension_start_code_identifier は、どの拡張データが送られるかを示すコードである。f_code[0][0] は、フォワード方向の水平動きベクトル探索範囲を表すデータである。f_code[0][1] は、フォワード方向の垂直動きベクトル探索範囲を表すデータである。f_code[1][0] は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f_code[1][1] は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。

【0472】 intra_dc_precision は、DC係数の精度を表すデータである。picture_structure は、フレームストラクチャかフィールドストラクチャかを示すデータである。フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示すデータである。top_field_first は、フレームストラクチャの場合、最初のフィールドが上位か下位かを示すデータである。frame_predictive_frame_dct は、フレーム・ストラクチャの場合、フレーム・モード DCT の予測がフレーム・モードだけであることを示すデータである。concealment_motion_vectors は、イントラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。

【0473】 q_scale_type は、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra_vlc_format は、イントラマクロブロックに、別の 2 次元 VLC を使うかどうかを示すデータである。alternate_scan は、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデータである。repeat_firt_field は、2 : 3 ブルダウンの際に使われるデータである。chroma_420_type は、信号フォーマットが 4 : 2 : 0 の場合、次の progressive_frame と同じ値、そうでない場合は 0 を表すデータである。progressive_frame は、このピクチャが、順次走査できているかどうかを示すデータである。composite_display_flag は、ソース信号がコンポジット信号であったかどうかを示すデータである。

【0474】 v_axis は、ソース信号が、PAL の場合に使われるデータである。field_sequence は、ソース信号が、PAL の場合に使われるデータである。sub_carrier は、ソース信号が、PAL の場合に使われるデータである。burst_amplitude は、ソース信号が、PAL の場合に使われるデータである。sub_carrier_phase は、ソース信号が、PAL の場合に使われるデータである。

【0475】 続いて、最終段の符号化処理によって生成

されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用された量子化マトリックスエクステンションが、履歴ストリームとして記述されている。

【0476】 この量子化マトリックスエクステンション (quant_matrix_extension) に関するデータエレメントは、extension_start_code, extension_start_code_identifier, quant_matrix_extension_present_flag, load_intra_quantiser_matrix, intra_quantiser_matrix[64], load_non_intra_quantiser_matrix, non_intra_quantiser_matrix[64], load_chroma_intra_quantiser_matrix, chroma_intra_quantiser_matrix[64], load_chroma_non_intra_quantiser_matrix, 及び chroma_non_intra_quantiser_matrix[64] から構成される。

【0477】 extension_start_code は、この量子化マトリックスエクステンションのスタートを示す開始コードである。extension_start_code_identifier は、どの拡張データが送られるかを示すコードである。quant_matrix_extension_present_flag は、この量子化マトリックスエクステンション内のデータエレメントが有効か無効かを示すためのデータである。load_intra_quantiser_matrix は、イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。intra_quantiser_matrix は、イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0478】 load_non_intra_quantiser_matrix は、非イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。non_intra_quantiser_matrix は、非イントラマクロブロック用の量子化マトリックスの値を表すデータである。load_chroma_intra_quantiser_matrix は、色差イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma_intra_quantiser_matrix は、色差イントラマクロブロック用の量子化マトリックスの値を示すデータである。load_chroma_non_intra_quantiser_matrix は、色差非イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma_non_intra_quantiser_matrix は、色差非イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0479】 続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたコピーライトエクステンションが、履歴ストリームとして記述されている。

【0480】 このコピーライトエクステンション (copyright_extension) に関するデータエレメントは、extension_start_code, extension_start_code_identifier, copyright_extension_present_flag, copyright_flag, copyright_identifier, original_or_copy, copyright_number_1, copyright_number_2, 及び copyright_number

r_3から構成される。

【0481】extension_start_codeは、コピーライトエクステンションのスタートを示す開始コードである。extension_start_code_identifierのどのエクステンションデータが送られるかを示すコードである。copyright_extension_present_flagは、このコピーライトエクステンション内のデータエレメントが有効か無効かを示すためのデータである。copyright_flagは、次のコピーライトエクステンション又はシーケンスエンドまで、符号化されたビデオデータに対してコピー権が与えられているか否かを示す。

【0482】copyright_identifierは、ISO/IEC JTC/SC 29によって指定されたコピー権の登録機関を識別するためのデータである。original_or_copyは、ビットストリーム中のデータが、オリジナルデータであるかコピーデータであるかを示すデータである。copyright_number_1は、コピーライトナンバーのビット44から63を表わすデータである。copyright_number_2は、コピーライトナンバーのビット22から43を表わすデータである。copyright_number_3は、コピーライトナンバーのビット0から21を表わすデータである。

【0483】続いて、最終段の符号化処理によって生成されたビットストリームのピクチャ層のユーザエリアには、過去の符号化処理において使用されたピクチャディスプレイエクステンション (picture_display_extension) が、履歴ストリームとして記述されている。

【0484】このピクチャディスプレイエクステンションを表わすデータエレメントは、extension_start_code、extension_start_code_identifier、picture_display_extension_present_flag、frame_center_horizontal_offset_1、frame_center_vertical_offset_1、frame_center_horizontal_offset_2、frame_center_vertical_offset_2、frame_center_horizontal_offset_3、及びframe_center_vertical_offset_3から構成される。

【0485】extension_start_codeは、ピクチャディスプレイエクステンションのスタートを示すための開始コードである。extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。picture_display_extension_present_flagは、ピクチャディスプレイエクステンション内のデータエレメントが有効か無効かを示すデータである。frame_center_horizontal_offsetは、表示エリアの水平方向のオフセットを示すデータであって、3つのオフセット値まで定義することができる。frame_center_vertical_offsetは、表示エリアを垂直方向のオフセットを示すデータであって、3つのオフセット値まで定義することができる。

【0486】最終段の符号化処理において生成されたビットストリームのピクチャ層のユーザエリアには、既に説明したピクチャディスプレイエクステンションを表わす履歴情報の次に、過去の符号化処理において使用され

たユーザデータ (user_data) が、履歴ストリームとして記述されている。

【0487】このユーザデータの次には、過去の符号化処理において使用されたマクロブロック層に関する情報が、履歴ストリームとして記述されている。

【0488】このマクロブロック層に関する情報は、macroblock_address_h、macroblock_address_v、slice_header_present_flag、skipped_macroblock_flag等のマクロブロック (macroblock) の位置に関するデータエレメントと、macroblock_quant、macroblock_motion_forward、macroblock_motion_backward、macroblock_pattern、macroblock_intra、spatial_temporal_weight_code_flag、frame_motion_type、及びdct_type等のマクロブロックモード (macroblock_modes[]) に関するデータエレメントと、quantiser_scale_code等の量子化ステップ制御に関するデータエレメントと、PMV[0][0][0]、PMV[0][0][1]、motion_vertical_field_select[0][0]、PMV[0][1][0]、PMV[0][1][1]、motion_vertical_field_select[0][1]、PMV[1][0][0]、PMV[1][0][1]、motion_vertical_field_select[1][0]、PMV[1][1][0]、PMV[1][1][1]、motion_vertical_field_select[1][1]等の動き補償に関するデータエレメントと、coded_block_pattern等のマクロブロックパターンに関するデータエレメントと、num_mv_bits、num_coef_bits、及びnum_other_bits等の発生符号量に関するデータエレメントから構成されている。

【0489】以下にマクロブロック層に関するデータエレメントについて詳細に説明する。

【0490】macroblock_address_hは、現在のマクロブロックの水平方向の絶対位置を定義するためのデータである。macroblock_address_vは、現在のマクロブロックの垂直方向の絶対位置を定義するためのデータである。slice_header_present_flagは、このマクロブロックがスライス層の先頭であり、スライスヘッダを伴うか否かを示すデータである。skipped_macroblock_flagは、復号処理においてこのマクロブロックをスキップするか否かを示すデータである。

【0491】macroblock_quantは、後述する図63と図64に示されたマクロブロックタイプ (macroblock_type) から導かれるデータであって、quantiser_scale_codeがビットストリーム中に現れるか否かを示すデータである。macroblock_motion_forwardは、図63と図64に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用されるデータである。macroblock_motion_backwardは、図63と図64に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用されるデータである。macroblock_patternは、図63と図64に示されたマクロブロックタイプから導かれるデータであって、coded_block_patternがビットストリーム中に現れるか否かを示すデータである。

【0492】macroblock_intraは、図63と図64に示されたマクロブロックタイプから導かれるデータであって、復号処理で使用するデータである。spatial_temporal_weight_code_flagは、図63と図64に示されたマクロブロックタイプから導かれるデータであって、時間スケラビリティで下位レイヤ画像のアップサンプリング方法を示すspatial_temporal_weight_codeは、ビットストリーム中に存在するか否かを示すデータである。

【0493】frame_motion_typeは、フレームのマクロブロックの予測タイプを示す2ビットのコードである。予測ベクトルが2個でフィールドベースの予測タイプであれば「00」であって、予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが1個でフレームベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプライムの予測タイプであれば「11」である。field_motion_typeは、フィールドのマクロブロックの動き予測を示す2ビットのコードである。予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが2個で18×8マクロブロックベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプライムの予測タイプであれば「11」である。dct_typeは、DCTがフレームDCTモードか、フィールドDCTモードかを示すデータである。quantiser_scale_codeはマクロブロックの量子化ステップサイズを示すデータである。

【0494】次に動きベクトルに関するデータエレメントについて説明する。動きベクトルは、復号時に必要な動きベクトルを減少させるために、先に符号化されたベクトルに関し差分として符号化される。動きベクトルの復号を行うために復号器は、4個の動きベクトル予測値（それぞれ水平及び垂直成分を伴う）を維持しなければいけない。この予測動きベクトルをPMV[r][s][v]と表わすことにしている。[r]は、マクロブロックにおける動きベクトルが第1のベクトルであるのか、第2のベクトルであるのかを示すフラグであって、マクロブロックにおけるベクトルが第1のベクトルである場合には「0」となって、マクロブロックにおけるベクトルが第2のベクトルである場合には「1」となる。[s]は、マクロブロックにおける動きベクトルの方向が、前方向であるのか後方向であるのかを示すフラグであって、前方向動きベクトルの場合には「0」となって、後方向動きベクトルの場合には「1」となる。[v]は、マクロブロックにおけるベクトルの成分が、水平方向であるのか垂直方向であるのかを示すフラグであって、水平方向成分の場合には「0」となって、垂直方向成分の場合には「1」となる。

【0495】従って、PMV[0][0][0]は、第1のベクトルの前方向の動きベクトルの水平方向成分のデータを表わし、PMV[0][0][1]は、第1のベクトルの前方向の動きベ

クトルの垂直方向成分のデータを表わし、PMV[0][1][0]は、第1のベクトルの後方向の動きベクトルの水平方向成分のデータを表わし、PMV[0][1][1]は、第1のベクトルの後方向の動きベクトルの垂直方向成分のデータを表わし、PMV[1][0][0]は、第2のベクトルの前方向の動きベクトルの水平方向成分のデータを表わし、PMV[1][0][1]は、第2のベクトルの前方向の動きベクトルの垂直方向成分のデータを表わし、PMV[1][1][0]は、第2のベクトルの後方向の動きベクトルの水平方向成分のデータを表わし、PMV[1][1][1]は、第2のベクトルの後方向の動きベクトルの垂直方向成分のデータを表わしている。

【0496】motion_vertical_field_select[r][s]は、予測の形式にいずれの参照フィールドを使用するかを示すデータである。このmotion_vertical_field_select[r][s]が「0」の場合には、トップ参照フィールドを使用し、「1」の場合には、ボトム参照フィールドを使用することを示している。

【0497】よって、motion_vertical_field_select[0][0]は、第1のベクトルの前方向の動きベクトルを生成する際の参照フィールドを示し、motion_vertical_field_select[0][1]は、第1のベクトルの後方向の動きベクトルを生成する際の参照フィールドを示し、motion_vertical_field_select[1][0]は、第2のベクトルの前方向の動きベクトルを生成する際の参照フィールドを示し、motion_vertical_field_select[1][1]は、第2ベクトルの後方向の動きベクトルを生成する際の参照フィールドを示している。

【0498】coded_block_patternは、DCT係数を格納する複数のDCTブロックのうち、どのDCTブロックに、有意係数（非0係数）があるかを示す可変長のデータである。num_mv_bitsは、マクロブロック中の動きベクトルの符号量を示すデータである。num_coef_bitsは、マクロブロック中のDCT係数の符号量を示すデータである。num_other_bitsは、マクロブロックの符号量で、動きベクトル及びDCT係数以外の符号量を示すデータである。

【0499】次に、可変長の履歴ストリームから各データエレメントをデコードするためのシグナックスについて、図47乃至図67を参照して説明する。

【0500】この可変長の履歴ストリームは、next_start_code()関数、sequence_header()関数、sequence_extension()関数、extension_and_user_data(0)関数、group_of_picture_header()関数、extension_and_user_data(1)関数、picture_header()関数、picture_coding_extension()関数、re_coding_stream_info()関数、extension_and_user_data(2)関数、及びpicture_data()関数によって定義されたデータエレメントによって構成される。

【0501】next_start_code()関数は、ビットストリーム中に存在するスタートコードを探すための関数であるので、履歴ストリームの最も先頭には、図48に示す

ような、過去の符号化処理において使用されたデータエレメントであってsequence_header()関数によって定義されたデータエレメントが記述されている。

【0502】sequence_header()関数によって定義されたデータエレメントは、sequence_header_code、sequence_header_present_flag、horizontal_size_value、vertical_size_value、aspect_ratio_information、frame_rate_code、bit_rate_value、marker_bit、VBV_buffer_size_value、constrained_parameter_flag、load_intra_quantiser_matrix、intra_quantiser_matrix、load_non_intra_quantiser_matrix、及びnon_intra_quantiser_matrix等である。

【0503】sequence_header_codeは、シーケンス層のスタート同期コードを表すデータである。sequence_header_present_flagは、sequence_header内のデータが有効か無効かを示すデータである。horizontal_size_valueは、画像の水平方向の画素数の下位12ビットから成るデータである。vertical_size_valueは、画像の縦のライン数の下位12ビットからなるデータである。aspect_ratio_informationは、画素のアスペクト比（縦横比）または表示画面アスペクト比を表すデータである。frame_rate_codeは、画像の表示周期を表すデータである。bit_rate_valueは、発生ビット量に対する制限のためのビット・レートの下位18ビット(400bsp単位で切り上げる)データである。

【0504】marker_bitは、スタートコードエミュレーションを防止するために挿入されるビットデータである。VBV_buffer_size_valueは、発生符号量制御用の仮想バッファ（ビデオバッファペリファイヤー）の大きさを決める値の下位10ビットデータである。constrained_parameter_flagは、各パラメータが制限以内であることを示すデータである。load_intra_quantiser_matrixは、イントラMB用量子化マトリックス・データの存在を示すデータである。intra_quantiser_matrixは、イントラMB用量子化マトリックスの値を示すデータである。load_non_intra_quantiser_matrixは、非イントラMB用量子化マトリックス・データの存在を示すデータである。non_intra_quantiser_matrixは、非イントラMB用量子化マトリックスの値を表すデータである。

【0505】sequence_header()関数によって定義されたデータエレメントの次には、図49で示すような、sequence_extension()関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。

【0506】sequence_extension()関数によって定義されたデータエレメントとは、extension_start_code、extension_start_code_identifier、sequence_extension_present_flag、profile_and_level_indication、progressive_sequence、chroma_format、horizontal_size_extension、vertical_size_extension、bit_rate_extension、vbv_buffer_size_extension、low_delay、frame_rate

e_extension_n、及びframe_rate_extension_d等のデータエレメントである。

【0507】extension_start_codeは、エクステンションデータのスタート同期コードを表すデータである。extension_start_code_identifierは、どの拡張データが送られるかを示すデータである。sequence_extension_present_flagは、シーケンスエクステンション内のデータが有効であるか無効であるかを示すデータである。profile_and_level_indicationは、ビデオデータのプロファイルとレベルを指定するためのデータである。progressive_sequenceは、ビデオデータが順次走査であることを示すデータである。chroma_formatは、ビデオデータの色差フォーマットを指定するためのデータである。horizontal_size_extensionは、シーケンスヘッダのhorizontal_size_valueに加える上位2ビットのデータである。vertical_size_extensionは、シーケンスヘッダのvertical_size_valueに加える上位2ビットのデータである。bit_rate_extensionは、シーケンスヘッダのbit_rate_valueに加える上位12ビットのデータである。vbv_buffer_size_extensionは、シーケンスヘッダのvbv_buffer_size_valueに加える上位8ビットのデータである。

【0508】low_delayは、Bピクチャを含まないことを示すデータである。frame_rate_extension_nは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを取得するためのデータである。frame_rate_extension_dは、シーケンスヘッダのframe_rate_codeと組み合わせてフレームレートを取得するためのデータである。

【0509】sequence_extension()関数によって定義されたデータエレメントの次には、図50に示すようなextension_and_user_data()関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。extension_and_user_data(i)関数は、「i」が1以外のときは、extension_data()関数によって定義されるデータエレメントは記述せずに、user_data()関数によって定義されるデータエレメントのみを履歴ストリームとして記述する。よって、extension_and_user_data(0)関数は、user_data()関数によって定義されるデータエレメントのみを履歴ストリームとして記述する。

【0510】user_data()関数は、図51に示されたようなシンタックスに基いて、ユーザデータを履歴ストリームとして記述する。

【0511】extension_and_user_data(0)関数によって定義されたデータエレメントの次には、図52に示すようなgroup_of_picture_header()関数によって定義されたデータエレメント、及びextension_and_user_data(1)関数によって定義されるデータエレメントが、履歴ストリームとして記述されている。但し、履歴ストリーム中に、GOP層のスタートコードを示すgroup_start_codeが記述されている場合にのみ、group_of_picture_header()関数によって定義されたデータエレメント、及びexte

nsion_and_user_data(1)関数によって定義されるデータエレメントが記述されている。

【0512】group_of_picture_header()関数によって定義されたデータエレメントは、group_start_code、group_of_picture_header_present_flag、time_code、closed_gop、及びbroken_linkから構成される。

【0513】group_start_codeは、GOP層の開始同期コードを示すデータである。group_of_picture_header_present_flagは、group_of_picture_header内のデータエレメントが有効であるか無効であることを示すデータである。time_codeは、GOPの先頭ピクチャのシーケンスの先頭からの時間を示すタイムコードである。closed_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

【0514】extension_and_user_data(1)関数は、extension_and_user_data(0)関数と同じように、user_data()関数によって定義されるデータエレメントのみを履歴ストリームとして記述する。

【0515】もし、履歴ストリーム中に、GOP層のスタートコードを示すgroup_start_codeが存在しない場合には、これらのgroup_of_picture_header()関数及びextension_and_user_data(1)関数によって定義されるデータエレメントは、履歴ストリーム中には記述されていない。その場合には、extension_and_user_data(0)関数によって定義されたデータエレメントの次に、picture_headr()関数によって定義されたデータエレメントが履歴ストリームとして記述されている。

【0516】picture_headr()関数によって定義されたデータエレメントは、図5.3に示すように、picture_start_code、temporal_reference、picture_coding_type、vbv_delay、full_pel_forward_vector、forward_f_code、full_pel_backward_vector、backward_f_code、extra_bit_picture、及びextra_information_pictureである。

【0517】具体的には、picture_start_codeは、ピクチャ層の開始同期コードを表すデータである。temporal_referenceは、ピクチャの表示順を示す番号でGOPの先頭でリセットされるデータである。picture_coding_typeは、ピクチャタイプを示すデータである。vbv_delayは、ランダムアクセス時の仮想バッファの初期状態を示すデータである。full_pel_forward_vectorは、順方向動きベクトルの精度が整数単位か半画素単位かを表すデータである。forward_f_codeは、順方向動きベクトル探索範囲を表すデータである。full_pel_backward_vectorは、逆方向動きベクトルの精度が整数単位か半画素単位かを表すデータである。backward_f_codeは、逆方向動きベクトル探索範囲を表すデータである。extra_bit_pictureは、後続する追加情報の存在を示すフラグであ

る。このextra_bit_pictureが「1」の場合には、次にextra_information_pictureが存在し、extra_bit_pictureが「0」の場合には、これに続くデータが無いことを示している。extra_information_pictureは、規格において予約された情報である。

【0518】picture_headr()関数によって定義されたデータエレメントの次には、図5.4に示すようなpicture_coding_extension()関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。

【0519】このpicture_coding_extension()関数によって定義されたデータエレメントとは、extension_start_code、extension_start_code_identifier、f_code[0][0]、f_code[0][1]、f_code[1][0]、f_code[1][1]、intra_dc_precision、picture_structure、top_field_first、frame_predictive_frame_dct、concealment_motion_vectors、q_scale_type、intra_vlc_format、alternate_scan、repeat_first_field、chroma_420_type、progressive_frame、composite_display_flag、v_axis、field_sequence、sub_carrier、burst_amplitude、及びsub_carrier_phaseから構成される。

【0520】extension_start_codeは、ピクチャ層のエクステンションデータのスタートを示す開始コードである。extension_start_code_identifierは、どの拡張データが送られるかを表すコードである。f_code[0][0]は、フォワード方向の水平動きベクトル探索範囲を表すデータである。f_code[0][1]は、フォワード方向の垂直動きベクトル探索範囲を表すデータである。f_code[1][0]は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f_code[1][1]は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。intra_dc_precisionは、DC係数の精度を表すデータである。

【0521】picture_structureは、フレームストラクチャかフィールドストラクチャかを表すデータである。フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示すデータである。top_field_firstは、フレームストラクチャの場合、最初のフィールドが上位か下位かを表すデータである。frame_predictive_frame_dctは、フレーム・ストラクチャの場合、フレーム・モードDCTの予測がフレーム・モードだけであることを示すデータである。concealment_motion_vectorsは、イントラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。q_scale_typeは、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra_vlc_formatは、イントラマクロブロックに、別の2次元VLCを使うかどうかを示すデータである。

【0522】alternate_scanは、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデ

ータである。repeat_firt_fieldは、2 : 3 プルダウンの際に使われるデータである。chroma_420_typeは、信号フォーマットが4 : 2 : 0の場合、次のprogressive_frameと同じ値、そうでない場合は0を表すデータである。progressive_frameは、このピクチャが、順次走査できているかどうかを示すデータである。composite_display_flagは、ソース信号がコンポジット信号であったかどうかを示すデータである。v_axisは、ソース信号が、PALの場合に使われるデータである。field_sequenceは、ソース信号が、PALの場合に使われるデータである。sub_carrierは、ソース信号が、PALの場合に使われるデータである。burst_amplitudeは、ソース信号が、PALの場合に使われるデータである。sub_carrier_phaseは、ソース信号が、PALの場合に使われるデータである。

【0523】 picture_coding_extension()関数によって定義されたデータエレメントの次には、re_coding_stream_info()関数によって定義されたデータエレメントが履歴ストリームとして記述されている。このre_coding_stream_info()関数は、主に履歴情報の組み合わせを記述する場合に用いられるものであり、その詳細については、図71を参照して後述する。

【0524】 re_coding_stream_info()関数によって定義されたデータエレメントの次には、extensions_and_user_data(2)によって定義されたデータエレメントが、履歴ストリームとして記述されている。このextension_and_user_data(2)関数は、図50に示したように、ビットストリーム中にエクステンションスタートコード(extension_start_code)が存在する場合には、extension_data()関数によって定義されるデータエレメントが記述されている。このデータエレメントの次には、ビットストリーム中にユーザデータスタートコード(user_data_start_code)が存在する場合には、user_data()関数によって定義されるデータエレメントが記述されている。但し、ビットストリーム中にエクステンションスタートコード及びユーザデータスタートコードが存在しない場合には extension_data()関数 及びuser_data()関数によって定義されるデータエレメントはビットストリーム中には記述されていない。

【0525】 extension_data()関数は、図55に示すように、extension_start_codeを示すデータエレメントと、quant_matrix_extension()関数、copyright_extension()関数、及びpicture_display_extension()関数によって定義されるデータエレメントとを、ビットストリーム中に履歴ストリームとして記述するための関数である。

【0526】 quant_matrix_extension()関数によって定義されるデータエレメントは、図56に示すように、extension_start_code、extension_start_code_identifier、quant_matrix_extension_present_flag、load_intra

_quantiser_matrix、intra_quantiser_matrix[64]、load_non_intra_quantiser_matrix、non_intra_quantiser_matrix[64]、load_chroma_intra_quantiser_matrix、chroma_intra_quantiser_matrix[64]、load_chroma_non_intra_quantiser_matrix、及びchroma_non_intra_quantiser_matrix[64] である。

【0527】 extension_start_codeは、この量子化マトリックスエクステンションのスタートを示す開始コードである。extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。quant_matrix_extension_present_flagは、この量子化マトリックスエクステンション内のデータエレメントが有効か無効かを示すためのデータである。load_intra_quantiser_matrixは、イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。intra_quantiser_matrixは、イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0528】 load_non_intra_quantiser_matrixは、非イントラマクロブロック用の量子化マトリックスデータの存在を示すデータである。non_intra_quantiser_matrixは、非イントラマクロブロック用の量子化マトリックスの値を表すデータである。load_chroma_intra_quantiser_matrixは、色差イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma_intra_quantiser_matrixは、色差イントラマクロブロック用の量子化マトリックスの値を示すデータである。load_chroma_non_intra_quantiser_matrixは、色差非イントラマクロブロック用の量子化マトリックス・データの存在を示すデータである。chroma_non_intra_quantiser_matrixは、色差非イントラマクロブロック用の量子化マトリックスの値を示すデータである。

【0529】 copyright_extension()関数によって定義されるデータエレメントは、図57に示すように、extension_start_code、extension_start_code_identifier、copyright_extension_present_flag、copyright_flag、copyright_identifier、original_or_copy、copyright_number_1、copyright_number_2、及び copyright_number_3から構成される。

【0530】 extension_start_codeは、コピーライトエクステンションのスタートを示す開始コードである。extension_start_code_identifierどのエクステンションデータが送られるかを示すコードである。copyright_extension_present_flagは、このコピーライトエクステンション内のデータエレメントが有効か無効かを示すためのデータである。

【0531】 copyright_flagは、次のコピーライトエクステンション又はシーケンスエンドまで、符号化されたビデオデータに対してコピー権が与えられているか否かを示す。copyright_identifierは、ISO/IEC JTC/SC29によって指定されたコピー権の登録機関を識別するための

データである。original_or_copyは、ビットストリーム中のデータが、オリジナルデータであるかコピーデータであることを示すデータである。copyright_number_1は、コピーライトナンバーのビット44から63を表わすデータである。copyright_number_2は、コピーライトナンバーのビット22から43を表わすデータである。copyright_number_3は、コピーライトナンバーのビット0から21を表わすデータである。

【0532】picture_display_extension0関数によって定義されるデータエレメントは、図58に示すように、extension_start_code_identifier、frame_center_horizontal_offset、frame_center_vertical_offset等である。

【0533】extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。frame_center_horizontal_offsetは、表示エリアの水平方向のオフセットを示すデータであって、number_of_frame_center_offsetsによって定義される数のオフセット値を定義することができる。frame_center_vertical_offsetは、表示エリアを垂直方向のオフセットを示すデータであって、number_of_frame_center_offsetsによって定義される数のオフセット値を定義することができる。

【0534】再び図47に戻って、extension_and_user_data(2)関数によって定義されるデータエレメントの次には、picture_data0関数によって定義されるデータエレメントが、履歴ストリームとして記述されている。但し、このpicture_data0関数は、red_bw_flagが1ではないか、または、red_bw_indicatorが2以下である場合に存在する。このred_bw_flagとred_bw_indicatorは、re_coding_stream_info0関数に記述されており、これらについては、図71と図72を参照して後述する。

【0535】picture_data0関数によって定義されるデータエレメントは、図59に示すように、slice0関数によって定義されるデータエレメントである。このslice0関数によって定義されるデータエレメントはビットストリーム中に少なくとも1個記述されている。

【0536】slice0関数は、図60に示されるように、slice_start_code、slice_quantiser_scale_code、intra_slice_flag、intra_slice、reserved_bits、extra_bit_slice、extra_information_slice、及びextra_bit_slice等のデータエレメントと、macroblock0関数によって定義されるデータエレメントを、履歴ストリームとして記述するための関数である。

【0537】slice_start_codeは、slice0関数によって定義されるデータエレメントのスタートを示すスタートコードである。slice_quantiser_scale_codeは、このスライス層に存在するマクロブロックに対して設定された量子化ステップサイズを示すデータである。しかし、各マクロブロック毎に、quantiser_scale_codeが設定されている場合には、各マクロブロックに対して設定され

たmacroblock_quantiser_scale_codeのデータが優先して使用される。

【0538】intra_slice_flagは、ビットストリーム中にintra_slice及びreserved_bitsが存在するか否かを示すフラグである。intra_sliceは、スライス層中にノンイントラマクロブロックが存在するか否かを示すデータである。スライス層におけるマクロブロックのいずれかがノンイントラマクロブロックである場合には、intra_sliceは「0」となり、スライス層におけるマクロブロックの全てがノンイントラマクロブロックである場合には、intra_sliceは「1」となる。reserved_bitsは、7ビットのデータであって「0」の値を取る。extra_bit_sliceは、履歴ストリームとして追加の情報が存在することを示すフラグであって、次にextra_information_sliceが存在する場合には「1」に設定される。追加の情報が存在しない場合には「0」に設定される。

【0539】これらのデータエレメントの次には、macroblock0関数によって定義されたデータエレメントが、履歴ストリームとして記述されている。

【0540】macroblock0関数は、図61に示すように、macroblock_escape、macroblock_address_increment、及びmacroblock_quantiser_scale_code、及びmarker_bit等のデータエレメントと、macroblock_modes0関数、motion_vectors(s)関数、及びcode_block_pattern0関数によって定義されたデータエレメントを記述するための関数である。

【0541】macroblock_escapeは、参照マクロブロックと前のマクロブロックとの水平方向の差が34以上であるか否かを示す固定ビット列である。参照マクロブロックと前のマクロブロックとの水平方向の差が34以上の場合には、macroblock_address_incrementの値に33をプラスする。macroblock_address_incrementは、参照マクロブロックと前のマクロブロックとの水平方向の差を示すデータである。もし、このmacroblock_address_incrementの前にmacroblock_escapeが1つ存在するのであれば、このmacroblock_address_incrementの値に33をプラスした値が、実際の参照マクロブロックと前のマクロブロックとの水平方向の差分を示すデータとなる。

【0542】macroblock_quantiser_scale_codeは、各マクロブロック毎に設定された量子化ステップサイズであり、macroblock_quantが「1」のときだけ存在する。各スライス層には、スライス層の量子化ステップサイズを示すslice_quantiser_scale_codeが設定されているが、参照マクロブロックに対してmacroblock_quantiser_scale_codeが設定されている場合には、この量子化ステップサイズを選択する。

【0543】macroblock_address_incrementの次には、macroblock_modes0関数によって定義されるデータエレメントが記述されている。macroblock_modes0関数は、図62に示すように、macroblock_type、frame_motion_

10

20

30

40

50

type、field_motion_type、dct_type等のデータエレメントを、履歴ストリームとして記述するための関数である。

【0544】macroblock_typeは、マクロブロックの符号化タイプを示すデータである。その詳細は、図65乃至図67を参照して後述する。

【0545】もし、macroblock_motion_forward又はmacroblock_motion_backwardが「1」であり、ピクチャ構造がフレームであり、さらにframe_pred_frame_dctが「0」である場合には、macroblock_typeを表わすデータエレメントの次にframe_motion_typeを表わすデータエレメントが記述されている。尚、このframe_pred_frame_dctは、frame_motion_typeがビットストリーム中に存在するか否かを示すフラグである。

【0546】frame_motion_typeは、フレームのマクロブロックの予測タイプを示す2ビットのコードである。予測ベクトルが2個でフィールドベースの予測タイプであれば「00」であって、予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが1個でフレームベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプライムの予測タイプであれば「11」である。

【0547】frame_motion_typeを記述する条件が満足されない場合には、macroblock_typeを表わすデータエレメントの次にfield_motion_typeを表わすデータエレメントが記述されている。

【0548】field_motion_typeは、フィールドのマクロブロックの動き予測を示す2ビットのコードである。予測ベクトルが1個でフィールドベースの予測タイプであれば「01」であって、予測ベクトルが2個で18×8マクロブロックベースの予測タイプであれば「10」であって、予測ベクトルが1個でディアルプライムの予測タイプであれば「11」である。

【0549】もし、ピクチャ構造がフレームで、frame_pred_frame_dctがframe_motion_typeがビットストリーム中に存在することを示し、且つ、frame_pred_frame_dctがdct_typeがビットストリーム中に存在することを示している場合には、macroblock_typeを表わすデータエレメントの次にdct_typeを表わすデータエレメントが記述されている。尚、dct_typeは、DCTがフレームDCTモードか、フィールドDCTモードかを示すデータである。

【0550】再び図61に戻って、もし、参照マクロブロックが前方予測マクロブロックであるか、又は参照マクロブロックがイントラマクロブロックであって且つコンシール処理のマクロブロックのいずれかの場合には、motion_vectors(0)関数によって定義されるデータエレメントが記述される。また、参照マクロブロックが後方予測マクロブロックである場合には、motion_vectors(1)関数によって定義されるデータエレメントが記述される。尚、motion_vectors(0)関数は、第1番目の動き

ベクトルに関するデータエレメントを記述するための関数であって、motion_vectors(1)関数は、第2番目の動きベクトルに関するデータエレメントを記述するための関数である。

【0551】motion_vectors(s)関数は、図63に示されるように、動きベクトルに関するデータエレメントを記述するための関数である。

【0552】もし、動きベクトルが1個でディアルプライム予測モードを使用していない場合には、motion_vertical_field_select[0][s]とmotion_vector(0,s)によって定義されるデータエレメントが記述される。

【0553】このmotion_vertical_field_select[r][s]は、第1番目の動きベクトル（前方又は後方のどちらのベクトルであっても良い）が、ボトムフィールドを参照して作られたベクトルであるかトップフィールドを参照して作られたベクトルであるかを示すフラグである。この指標“r”は、第1番目のベクトル又は第2番目のベクトルのいずれのベクトルであるかを示す指標であって、“s”は、予測方向が前方又は後方予測のいずれであるかを示す指標である。

【0554】motion_vector(r,s)関数は、図64に示されるように、motion_code[r][s][t]に関するデータ列と、motion_residual[r][s][t]に関するデータ列と、dmvector[t]を表わすデータとを記述するための関数である。

【0555】motion_code[r][s][t]は、動きベクトルの大きさを-16～+16の範囲で表わす可変長のデータである。motion_residual[r][s][t]は、動きベクトルの残差を表わす可変長のデータである。よって、このmotion_code[r][s][t]とmotion_residual[r][s][t]との値によって詳細な動きベクトルを記述することができる。dmvector[t]は、ディアルプライム予測モードのときに、一方のフィールド（例えばボトムフィールドに対してトップフィールドを一方のフィールドとする）における動きベクトルを生成するために、時間距離に応じた既存の動きベクトルがスケールされると共に、トップフィールドとボトムフィールドとのライン間の垂直方向のずれを反映させるために垂直方向に対して補正を行うデータである。この指標“r”は、第1番目のベクトル又は第2番目のベクトルのいずれのベクトルであるかを示す指標であって、“s”は、予測方向が前方又は後方予測のいずれであるかを示す指標である。“s”は、動きベクトルが垂直方向の成分であるか水平方向の成分であるかを示すデータである。

【0556】図64に示されmotion_vector(r,s)関数によって、まず、水平方向のmotion_coder[r][s][0]を表わすデータ列が、履歴ストリームとして記述される。motion_residual[0][s][t]及びmotion_residual[1][s][t]の双方のビット数は、f_code[s][t]で示されるので、f_code[s][t]が1でない場合には、motion_residual[r]

[s][t] がビットストリーム中に存在することを示すことになる。水平方向成分のmotion_residual[r][s][0]が「1」でなくて、水平方向成分のmotion_code[r][s][0]が「0」でないということは、ビットストリーム中にmotion_residual[r][s][0]を表わすデータエレメントが存在し、動きベクトルの水平方向成分が存在するということを意味している。その場合には、水平方向成分のmotion_residual[r][s][0]を表わすデータエレメントが記述されている。

【0557】続いて、垂直方向のmotion_coder[r][s][t]を表わすデータ列が、履歴ストリームとして記述される。同じようにmotion_residual[0][s][t]及びmotion_residual[t][s][t]の双方のビット数は、f_code[s][t]で示されるので、f_code[s][t]が1でない場合には、motion_residual[r][s][t]がビットストリーム中に存在することを表わすことになる。motion_residual[r][s][t]が「1」でなくて、motion_code[r][s][t]が「0」でないということは、ビットストリーム中にmotion_residual[r][s][t]を表わすデータエレメントが存在し、動きベクトルの垂直方向成分が存在するということを意味している。その場合には、垂直方向成分のmotion_residual[r][s][t]を表わすデータエレメントが記述されている。

【0558】次に、図65乃至図67を参照して、macroblock_typeについて説明する。macroblock_typeは、macroblock_quant、dct_type_flag、macroblock_motion_forward、及びmacroblock_motion_backwardなどのフラグから生成された可変長データである。macroblock_quantは、マクロブロックに対して量子化ステップサイズを設定するためのmacroblock_quantiser_scale_codeが設定されているか否かを示すフラグであって、ビットストリーム中にmacroblock_quantiser_scale_codeが存在する場合には、macroblock_quantは「1」の値を取る。

【0559】dct_type_flagは、参照マクロブロックがフレームDCT又はフィールドDCTで符号化されているかを示すdct_typeが存在するか否かを示すためのフラグ（言い換えるとDCTされているか否かを示すフラグ）であって、ビットストリーム中にdct_typeが存在する場合には、このdct_type_flagは「1」の値を取る。macroblock_motion_forwardは、参照マクロブロックが前方予測されているか否かを示すフラグであって、前方予測されている場合には「1」の値を取る。macroblock_motion_backwardは、参照マクロブロックが後方予測されているか否かを示すフラグであって、後方予測されている場合には「1」の値を取る。

【0560】なお、可変長フォーマットにおいては、伝送するビットレートを減少させるために、履歴情報を削減することができる。

【0561】すなわち、macroblock_typeとmotion_vectors()は転送するが、quantiser_scale_codeを転送しな

い場合には、slice_quantiser_scale_codeを”00000”とすることで、ビットレートを減少させることができる。

【0562】また、macroblock_typeのみ転送し、motion_vectors()、quantiser_scale_code、およびdct_typeを転送しない場合には、macroblock_typeとして、”not coded”を使用することで、ビットレートを減少することができる。

【0563】さらにまた、picture_coding_typeのみ転送し、slice()以下の情報は全て転送しない場合には、slice_start_codeを持たないpicture_data()を使用することで、ビットレートを減少させることができる。

【0564】以上においては、user_data内の23ビットの連続する”0”が出ないようにする場合に、22ビット毎に”1”を挿入するようにしたが、22ビット毎でなくてもよい。また、連続する”0”の個数を数えて”1”を挿入するのではなく、Byte_alignを調べて挿入するようにすることも可能である。

【0565】さらに、MPEGにおいては、23ビットの連続する”0”の発生を禁止しているが、実際には、バイトの先頭から23ビット連続する場合だけが問題とされ、バイトの先頭ではなく、途中から0が23ビット連続する場合は、問題とされない。従って、例えば24ビット毎に、LSB以外の位置に”1”を挿入するようにしてもよい。

【0566】また、以上においては、履歴情報を、video elementary streamに近い形式にしたが、packetized elementary streamやtransport streamに近い形式にしてもよい。また、Elementary Streamのuser_dataの場所を、picture_dataの前としたが、他の場所にしてもできる。

【0567】図15のトランスコード101においては、4世代分の符号化パラメータを履歴情報として後段に出力するようにしたが、実際には、履歴情報の全てが必要となるわけではなく、アプリケーション毎に必要な履歴情報は異なってくる。また、実際の伝送路あるいは記録媒体（伝送メディア）には、容量に制限があり、圧縮しているとはいえ、全ての履歴情報を伝送するようにすると、容量的に負担となり、結果的に画像ビットストリームのビットレートを抑圧してしまい、履歴情報伝送の有効性が損なわれることになる。

【0568】そこで、履歴情報として伝送する項目の組み合わせを記述する記述子を履歴情報に組み込んで後段に送信するようにし、全ての履歴情報を伝送するのではなく、様々なアプリケーションに対応した情報を伝送するようにすることができる。図68は、このような場合のトランスコード101の構成例を表している。

【0569】図68において、図15における場合と対応する部分には同一の符号を付してあり、その説明は適宜省略する。図68の構成例においては、ヒストリ情報

分離装置 105 と符号化装置 106 の間、及びヒストリエンコーディング装置 107 と符号化装置 106 の間に、符号化パラメータ選択回路 501 が挿入されている。

【0570】符号化パラメータ選択回路 501 は、ヒストリ情報分離装置 105 が出力するベースバンドビデオ信号から符号化パラメータを算出する符号化パラメータ算出部 512、ヒストリ情報分離装置 105 が出力する、このトランスコーダ 101 において、符号化するのに最適と判定された符号化パラメータ（例えば、第 2 世代の符号化パラメータ）に関する情報から、符号化パラメータと記述子（red_bw_flag, red_bw_indicator）

（図 72 を参照して後述する）を分離する組合せ記述子分離部 511、並びに符号化パラメータ算出部 512 が出力する符号化パラメータと、組合せ記述子分離部 511 が出力する符号化パラメータのうち、いずれか一方を、組合せ記述子分離部 511 で分離された記述子に対応して選択し、符号化装置 106 に出力するスイッチ 513 を有している。その他の構成は、図 15 における場合と同様である。

【0571】ここで、履歴情報として伝送する項目の組み合わせについて説明する。履歴情報は、分類すると、picture 単位の情報と、macroblock 単位の情報に分けることができる。slice 単位の情報は、それに含まれる macroblock の情報を収集することで得ることができ、GOP 単位の情報は、それに含まれる picture 単位の情報を収集することで得ることができる。

【0572】picture 単位の情報は、1 フレーム毎に 1 回伝送されるだけなので、情報伝送に占めるビットレートは、それほど大きくはない。これに対して、macroblock 単位の情報は、各 macroblock 毎に伝送されるため、例えば 1 フレームの走査線数が 525 本で、フィールドレートが 60 フィールド/秒のビデオシステムの場合、1 フレームの画素数を 720×480 とすると、macroblock 単位の情報は、1 フレームあたり $1350 \text{ (} = (720 / 16) \times (480 / 16) \text{)}$ 回伝送することが必要となる。このため、履歴情報の相当の部分が macroblock 毎の情報で占められることになる。そこで、履歴情報としては、少なくとも picture 単位の情報は常に伝送するが、macroblock 単位の情報は、アプリケーションに応じて選択して伝送するようにすることで、伝送する情報量を抑制することができる。

【0573】履歴情報として転送される macroblock 単位の情報には、例えば num_coef_bits, num_mv_bits, num_other_bits, q_scale_code, q_scale_type, motion_type, mv_vert_field_sel, mv, mb_mfwd, mb_mbw, mb_pattern, coded_block_pattern, mb_intra, slice_start, dct_type, mb_quant, skipped_mb などがある。これらは、macroblock rate information の要素を用いて表現されたものである。

【0574】num_coef_bits は、macroblock の符号量のうち、DCT 係数に要した符号量を表す。num_mv_bits は、macroblock の符号量のうち、動きベクトルに要した符号量を表す。num_other_bits は、macroblock の符号量のうち、num_coef_bits 及び num_mv_bits 以外の符号量を表す。

【0575】q_scale_code は、macroblock に適用された q_scale_code を表す。motion_type は、macroblock に適用された動きベクトルの type を表す。mv_vert_field_sel は、macroblock に適用された動きベクトルの field select を表す。

【0576】mv は、macroblock に適用された動きベクトルを表す。mb_mfwd は、macroblock の予測モードが前方向予測であることを示すフラグである。mb_mbw は、macroblock の予測モードが後方向予測であることを示すフラグである。mb_pattern は、macroblock の DCT 係数の非 0 のものの有無を示すフラグである。

【0577】coded_block_pattern は、macroblock の DCT 係数の非 0 のものの有無を DCT ブロック毎に示すフラグである。mb_intra は、macroblock が intra_macro かそうでないかを示すフラグである。slice_start は、macroblock が slice の先頭であるか否かを示すフラグである。dct_type は、macroblock が field_dct か frame_dct かを示すフラグである。

【0578】mb_quant は、macroblock が quantiser_scale_code を伝送するか否かを示すフラグである。skipped_mb は、macroblock が skipped macroblock であるか否かを示すフラグである。

【0579】これらの項目は、常に全て必要であるわけではなく、アプリケーションに応じて必要となる項目が変化する。例えば、num_coef_bits や slice_start といった項目は、再エンコードした際のビットストリームをできる限り元の形に戻したいという transparent という要求を有するアプリケーションにおいて必要となる。換言すれば、ビットレートを変更するようなアプリケーションにおいては、これらの項目は必要ではない。また、非常に伝送路の制限が厳しい場合には、各ピクチャの符号化タイプが判るだけでもよいようなアプリケーションも存在する。このような状況から、履歴情報を伝送する項目の組み合わせの例として、例えば図 69 に示すような組み合わせが考えられる。

【0580】図 69 において、各組み合わせの中の項目に対応する値「2」は、その情報が存在し、利用可能であることを意味し、「0」は、その情報が存在しないことを意味する。「1」は、他の情報の存在を補助する目的のため、あるいは、構文上存在するが、元のビットストリーム情報とは関係がないなど、その情報自身には意味がないことを表している。例えば、slice_start は、履歴情報を伝送する際の slice の先頭の macroblock において、「1」になるが、本来のビットストリームに対し

て、sliceが必ずしも同一位置関係にあるわけではない場合には、履歴情報としては無意味になる。

【0581】図69の例においては、(num_coef_bits, num_mv_bits, num_other_bits), (q_scale_code, q_scale_type), (motion_type, mv_vert_field_sel, mv, mb_mfwd, mb_mbwd), (mb_pattern), (coded_block_pattern), (mb_intra), (slice_start), (dct_type), (mb_quant), (skipped_mb)の各項目の有無により、組み合わせ1乃至組み合わせ5の5つの組み合わせが用意されている。

【0582】組み合わせ1は、完全にtransparentなビットストリームを再構成することを目的とした組み合わせである。この組み合わせによれば、発生符号量情報を用いることによる精度の高いトランスコーディングが実現できる。組み合わせ2も、完全にtransparentなビットストリームを再構成することを目的とした組み合わせである。組み合わせ3は、完全にtransparentなビットストリームを再構成することはできないが、視覚的にはtransparentなビットストリームを再構成できるようにするための組み合わせである。組み合わせ4は、transparentという観点からは組み合わせ3よりも劣るが、視覚上問題がないビットストリームの再構成ができる組み合わせである。組み合わせ5は、transparentという観点からは組み合わせ4よりも劣るが、少ない履歴情報でビットストリームの完全ではない再構成ができる組み合わせである。

【0583】これらの組み合わせのうち、組み合わせの番号の数字が小さいものほど、機能的には上位であるが、履歴を転送するのに必要となる容量が多くなる。従って、想定するアプリケーションと履歴に使用できる容量を考慮することによって、伝送する組み合わせを決定する必要がある。

【0584】次に、図70のフローチャートを参照して、図68のトランスコーダ101の動作について説明する。ステップS41において、トランスコーダ101の復号装置102は、入力されたビットストリームを復号し、そのビットストリームを符号化の際に使用された符号化パラメータ(4th)を抽出し、その符号化パラメータ(4th)を履歴情報多重化装置103に出力するとともに、復号したビデオデータをやはり履歴情報多重化装置103に出力する。ステップS42において、復号装置102はまた、入力されたビットストリームからuser_dataを抽出し、履歴デコーディング装置104に出力する。履歴デコーディング装置104は、ステップS43において、入力されたuser_dataから、組み合わせ情報(記述子)を抽出し、さらにそれを用いて、履歴情報としての符号化パラメータ(1st, 2nd, 3rd)を抽出し、履歴情報多重化装置103に出力する。

【0585】履歴情報多重化装置103は、ステッ

プS44において、ステップS41で取り出された復号装置102から供給される現在の符号化パラメータ(4th)と、ステップS43で履歴デコーディング装置104が出力した過去の符号化パラメータ(1st, 2nd, 3rd)とを、復号装置102から供給されるベースバンドのビデオデータに、図18または図31に示すようなフォーマットに従って多重化し、履歴情報分離装置105に出力する。

【0586】履歴情報分離装置105は、ステップS45において、履歴情報多重化装置103より供給されたベースバンドのビデオデータから符号化パラメータを抽出し、その中から今回の符号化に最も適している符号化パラメータ(例えば、第2世代の符号化パラメータ)を選択し、記述子とともに、組合せ記述子分離部511に出力する。また、履歴情報分離装置105は、今回の符号化に最適と判定された符号化パラメータ以外の符号化パラメータ(例えば、最適な符号化パラメータが第2世代の符号化パラメータであると判定された場合には、それ以外の第1世代、第3世代、及び第4世代の符号化パラメータ)を履歴エンコーディング装置107に出力する。履歴エンコーディング装置107は、履歴情報分離装置105より入力された符号化パラメータをステップS46において、user_dataに記述し、そのuser_data(converted_history_stream0)を符号化装置106に出力する。

【0587】符号化パラメータ選択回路501の組合せ記述子分離部511は、履歴情報分離装置105より供給されたデータから、符号化パラメータと記述子を分離し、符号化パラメータ(2nd)をスイッチ513の一方の接点に供給する。スイッチ513の他方の接点には、符号化パラメータ算出部512が、履歴情報分離装置105が出力するベースバンドのビデオデータから、符号化パラメータを算出し、供給している。スイッチ513は、ステップS48において、組合せ記述子分離部511が出力した記述子に対応して、組合せ記述子分離部511が出力した符号化パラメータ、または符号化パラメータ算出部512が出力した符号化パラメータのいずれかを選択し、符号化装置106に出力する。すなわち、スイッチ513では、組合せ記述子分離部511から供給された符号化パラメータが有効である場合には、組合せ記述子分離部511が出力する符号化パラメータが選択されるが、組合せ記述子分離部511が出力する符号化パラメータが無効であると判定された場合には、符号化パラメータ算出部512がベースバンドビデオを処理することで算出した符号化パラメータが選択される。この選択は、伝送メディアの容量に対応して行われる。

【0588】符号化装置106は、ステップS49において、スイッチ513から供給された符号化パラメータに基づいて、履歴情報分離装置105より供給され

たベースバンドビデオ信号を符号化する。また、ステップ S50において、符号化装置 106は、符号化したビットストリームに、ヒストリエコーディング装置 107より供給されたuser_dataを多重化し、出力する。

【0589】このようにして、各履歴によって得られる符号化パラメータの組み合わせが異なっているような場合でも、支障なくトランスコーディングすることが可能となる。

【0590】このように、履歴情報は、図 38に示したように、ビデオストリームのuser_data関数の一種としてのhistory_stream0（より正確には、converted_history_stream0）で伝送される。そのhistory_stream0のシンタックスは、図 47に示した通りである。履歴情報の項目の組み合わせを表す記述子（red_bw_flag, red_bw_indicator）、およびMPEGのストリームではサポートされていない項目（num_other_bits, num_mv_bits, num_coef_bits）は、この図 47の中のre_coding_stream_info0関数により伝送される。

【0591】re_coding_stream_info0関数は、図 71に示すように、user_data_start_code, re_coding_stream_info_ID, red_bw_flag, red_bw_indicator, marker_bit, num_other_bits, num_mv_bits, num_coef_bitsなどのデータエレメントより構成される。

【0592】user_data_start_codeは、user_dataが開始することを表すスタートコードである。re_coding_stream_info_IDは、16ビットの整数であり、re_coding_stream_info0関数の識別のために用いられる。その値は、具体的には、“1001 00011110 1100”（0x91ec）とされる。

【0593】red_bw_flagは、1ビットのフラグであり、履歴情報が全ての項目を伝送する場合には0とされ、このフラグの値が1である場合、このフラグに続くred_bw_indicatorを調べることにより、図 69に示した5個の組み合わせのうち、どの組み合わせで項目が送られているのかを決定することができる。

【0594】red_bw_indicatorは、2ビットの整数であり、項目の組み合わせを図 72に示すように記述する。

【0595】即ち、図 69に示した5つの組み合わせのうち、組み合わせ1の場合、red_bw_flagは0とされ、組み合わせ2乃至組み合わせ5のとき、red_bw_flagは1とされる。これに対して、red_bw_indicatorは、組み合わせ2の場合0とされ、組み合わせ3の場合1とされ、組み合わせ4の場合2とされ、組み合わせ5の場合3とされる。

【0596】従って、red_bw_indicatorは、red_bw_flagが1の場合に（組み合わせ2乃至組み合わせ5の場合に）規定される。

【0597】さらに、図 71に示すように、red_bw_flagが0である場合（組み合わせ1の場合）、マクロブロック毎に、marker_bit, num_other_bits, num_mv_bit

s, num_coef_bitsが記述される。これら4つのデータエレメントは、組み合わせ2乃至組み合わせ5の場合（red_bw_flagが1の場合）規定されない。

【0598】図 59に示したように、picture_data0関数は、1個以上のslice0関数から構成される。しかしながら、組み合わせ5の場合、picture_data0関数を含めて、それ以下のシンタックス要素は伝送されない（図 69）。この場合、履歴情報は、picture_typeなどのpicture単位の情報の伝送を意図したものとなる。

10 【0599】組み合わせ1乃至組み合わせ4の場合、図 60に示したslice0関数が存在する。しかしながら、このslice0関数によって決定されるsliceの位置情報と、元のビットストリームのsliceの位置情報は、履歴情報の項目の組み合わせに依存する。組み合わせ1または組み合わせ2の場合、履歴情報の元となったビットストリームのsliceの位置情報と、slice0関数によって決定されるsliceの位置情報とは、同一である必要がある。

20 【0600】図 61に示すmacroblock0関数のシンタックス要素は、履歴情報の項目の組み合わせに依存する。macroblock_escape, macroblock_address_increment, macroblock_modes0関数は、常に存在する。しかしながら、macroblock_escapeとmacroblock_address_incrementの情報としての有効性は、組み合わせによって決定される。履歴情報の項目の組み合わせが、組み合わせ1または組み合わせ2の場合、元のビットストリームのskippped_mb情報と同じものが伝送される必要がある。

30 【0601】組み合わせ4の場合、motion_vectors0関数は存在しない。組み合わせ1乃至組み合わせ3の場合、macroblock_modes0関数のmacroblock_typeによって、motion_vectors0関数の存在が決定される。組み合わせ3または組み合わせ4の場合には、coded_block_pattern0関数は存在しない。組み合わせ1と組み合わせ2の場合、macroblock_modes0関数のmacroblock_typeによって、coded_block_pattern0関数の存在が決定される。

40 【0602】図 62に示したmacroblock_modes0関数のシンタックス要素は、履歴情報の項目の組み合わせに依存する。macroblock_typeは、常に存在する。組み合わせが組み合わせ4である場合、flame_motion_type, field_motion_type, dct_typeは存在しない。

【0603】macroblock_typeより得られるパラメータの情報としての有効性は、履歴情報の項目の組み合わせによって決定される。

50 【0604】履歴情報の項目の組み合わせが組み合わせ1または組み合わせ2である場合、macroblock_quantは、元のビットストリームと同じである必要がある。組み合わせ3または組み合わせ4の場合、macroblock_quantは、macroblock0関数内のquantiser_scale_codeの存在を表し、元のビットストリームと同じである必要はな

い。

【0605】組み合わせが組み合わせ1乃至組み合わせ3である場合、macroblock_motion_forwardとmacroblock_motion_backwardは、元のビットストリームと同一である必要がある。組み合わせが組み合わせ4または組み合わせ5である場合、その必要はない。

【0606】組み合わせが組み合わせ1または組み合わせ2である場合、macroblock_patternは、元のビットストリームと同一である必要がある。組み合わせ3の場合、macroblock_patternは、dct_typeの存在を示すのに用いられる。組み合わせが組み合わせ4である場合、組み合わせ1乃至組み合わせ3における場合のような関係は成立しない。

【0607】履歴情報の項目の組み合わせが組み合わせ1乃至組み合わせ3の場合、macroblock_intraは、元のビットストリームと同一である必要がある。組み合わせ4の場合には、その限りでない。

【0608】図47のhistory_stream()は、履歴情報を可変長とする場合のシンタックスであるが、図40乃至図46に示すように、固定長のシンタックスとする場合、固定長の履歴情報内に、伝送される項目中のどれが有効であるかを示す情報としての記述子 (red_bw_flag と red_bw_indicator) をベースバンド画像に重畳し、伝送するようにする。その結果、この記述子を調べることにより、フィールドとして存在するが、その内容は無効であるといった判断をすることが可能となる。

【0609】このため、図44に示すように、re_coding_stream_informationとして、user_data_start_code, re_coding_stream_info_ID, red_bw_flag, red_bw_indicator, marker_bitが配置されている。それぞれの意味は、図71における場合と同様である。

【0610】このように履歴として伝送する符号化パラメータの要素をアプリケーションに応じた組み合わせで伝送するようにすることで、アプリケーションに応じた履歴を適当なデータ量で伝送するようにすることができる。

【0611】以上のように、履歴情報を可変長符号として伝送する場合、re_coding_stream_info()関数は、図71に示すように構成され、図47に示すように、history_stream()関数の一部として伝送される。これに対して、履歴情報を固定長符号として伝送する場合には、図44に示したように、history_stream()関数の一部として、re_coding_stream_information()が伝送される。図44の例では、re_coding_stream_informationとして、user_data_start_code, re_coding_stream_info_ID, red_bw_flag, red_bw_indicatorが伝送される。

【0612】また、図68のヒストリ情報多重化装置103が出力するベースバンドの信号中における履歴情報の伝送のために、図73に示すようなRe_Coding information Bus macroblock formatが規定される。このマク

ロブロックは、 $16 \times 16 (=256)$ ビットで構成される。そして、そのうちの図73において上から3行目と4行目に示す32ビットが、picrate_elementとされる。このpicrate_elementには、図74乃至図76に示すPicture rate elementsが記述される。図74の上から2行目に1ビットのred_bw_flagが規定されており、また、3行目に3ビットのred_bw_indicatorが規定されている。即ち、これらのフラグred_bw_flag, red_bw_indicatorは、図73のpicrate_elementとして伝送される。

【0613】図73のその他のデータについて説明すると、SRIB_sync_codeは、このフォーマットのマクロブロックの最初の行が左詰めにアライメントされていることを表すコードであり、具体的には、“11111”に設定される。fr_fl_SRIBは、picture_structureがフレームピクチャ構造の場合（その値が“11”である場合）、1に設定され、Re_Coding Information Bus macroblockが16ラインを超えて伝送されることを表し、picture_structureがフレーム構造ではない場合、0に設定され、Re_Coding Information Busが16ラインを超えて伝送されることを意味する。この機構により、Re_Coding Information Busが、空間的かつ時間的にデコードされたビデオフレームまたはフィールドの対応する画素にロックされる。

【0614】SRIB_top_field_firstは、元のビットストリームに保持されているtop_field_firstと同じ値に設定され、関連するビデオのRe_Coding Information Busの時間的アライメントをrepeat_first_fieldとともに表している。SRIB_repeat_first_fieldは、元のビットストリームに保持されているrepeat_first_fieldと同じ値に設定される。first fieldのRe_Coding Information Busの内容は、このフラグに示されるように繰り返される必要がある。

【0615】422_420_chromaは、元のビットストリームが4:2:2または4:2:0のいずれであるかを表す。その値の0は、ビットストリームが4:2:0であり、色差信号のアップサンプリングが、4:2:2のビデオが出力されるように行われたことを表す。その値の0は、色差信号のフィルタリング処理が実行されていないことを表す。

【0616】rolling_SRIB_mb_refは、16ビットのモジュロ65521を表し、この値は、毎マクロブロック毎にインクリメントされる。この値は、フレームピクチャ構造のフレームに渡って連続している必要がある。さもなくば、この値は、フィールドに渡って連続している必要がある。この値は、0から65520の間の所定の値に初期化される。これにより、レコーダのシステムに、ユニークなRe_Coding Information Busの識別子を組み込むことが許容される。

【0617】Re_Coding Information Bus macroblockの

その他のデータの意味は、上述した通りであるので、ここでは省略する。

【0618】図77に示すように、図73の256ビットのRe_Coding Information Busのデータは、1ビットずつ、色差データのLSBであるCb[0][0], Cr[0][0], Cb[1][0], Cr[1][0]に配置される。図77に示すフォーマットにより、4ビットのデータを送ることができるので、図73の256ビットのデータは、図77のフォーマットを64 (=256/4) 個送ることで伝送することができる。

【0619】本発明のトランスコードによれば、過去の符号化処理において生成された符号化パラメータを、現在の符号化処理において再利用するようにしているので、復号処理及び符号化処理を繰り返したとしても画質劣化が発生しない。つまり、復号処理及び符号化処理の繰り返しによる画質劣化の蓄積を低減することができる。

【0620】図78と図79は、本発明のトランスコードをビデオテーブルコードに適用した場合の構成例を表している。図78は、ビデオテーブルコード601の記録系の構成例を表しており、図79は、ビデオテーブルコード601の再生系の構成例を表している。

【0621】図78のビデオテーブルコード601は、トランスコード101R、チャンネルエンコーディング装置602、記録ヘッド603により構成されている。トランスコード101Rの構成は、図37に示したトランスコードと基本的に同様とされている。この構成例においては、トランスコード101Rにおいては、LongGOPのビットストリームSTが、Short GOPのビットストリームSTに変換される。

【0622】トランスコード101Rの符号化装置106より出力された、第4世代の符号化ストリームSTが、チャンネルエンコーディング装置602に供給される。上述したように、この第4世代の符号化ストリームSTのピクチャ層のユーザデータエリアには、第1世代乃至第3世代の符号化パラメータを含むユーザデータuser_dataが記録されている。

【0623】チャンネルエンコーディング装置602は、入力された第4世代の符号化ストリームに、誤り訂正のためのパリティ符号を付けた後、例えば、NRZI変調方式でチャンネルエンコードし、記録ヘッド603に供給する。記録ヘッド603は入力された符号化ストリームを磁気テープ604に記録する。

【0624】図79に示すように、再生系においては、磁気テープ604から、再生ヘッド611により信号が生成され、チャンネルデコーディング装置612に供給される。チャンネルデコーディング装置612は、再生ヘッド611から供給された信号をチャンネルデコードし、パリティを用いて、誤り訂正する。

【0625】チャンネルデコーディング装置612によ

り出力された第4世代の符号化ストリームSTは、トランスコード101Pに入力される。トランスコード101Pの基本的な構成は、図37に示したトランスコードと同様の構成とされている。

【0626】トランスコード101Pの復号装置102は、第4世代の符号化ストリームから、第1世代乃至第3世代の符号化パラメータを含むユーザデータuser_dataを抽出し、ヒストリデコーディング装置104と、符号化装置106に供給する。ヒストリデコーディング装置104は、入力されたユーザデータuser_dataを復号し、得られた第1世代乃至第3の符号化パラメータを符号化装置106に供給する。

【0627】復号装置102はまた、第4世代の符号化ストリームSTを復号し、ベースバンドビデオ信号と、第4世代の符号化パラメータを出力する。ベースバンドビデオ信号は、符号化装置106に供給され、第4世代の符号化パラメータは、符号化装置106と、ヒストリエンコーディング装置107に供給される。

【0628】ヒストリエンコーディング装置107は、入力された第4世代の符号化パラメータを、ユーザデータuser_dataに変換し、符号化装置106に供給する。

【0629】上述したように、符号化装置106のコントローラ70は、オペレータによって指定されたGOP構造から決定された各ピクチャのピクチャタイプと、ヒストリ情報（ユーザデータuser_data）に含まれるピクチャタイプが一致するか否かを判断する。そしてその判断結果に対応して、上述した、「通常符号化処理」、または「パラメータ再利用符号化処理」を実行する。この処理により符号化装置106から、Short GOPからLong GOPに変換された、第4世代の符号化ストリームSTが出力される。この符号化ストリームSTのユーザデータuser_dataには、第1世代乃至第4世代の符号化パラメータがヒストリ情報として記録されている。

【0630】図78と図79に示したビデオテーブルコード601においては、ヒストリ情報をピクチャレイヤのuser_dataに記録するようにしたが、ヒストリ情報は、磁気テープ604のビデオデータとは異なる領域に記録することも可能である。図80と図81は、この場合のビデオテーブルコード601の構成例を表している。図80は、ビデオテーブルコード601の記録系の構成例を表しており、図81は、再生系の構成例を表している。

【0631】図80に示すように、このビデオテーブルコード601においては、そのトランスコード101Rの復号装置102より出力されたユーザデータuser_dataが、ヒストリデコーディング装置104に入力され、そこで過去の符号化パラメータ（この例の場合、第1世代と第2世代の符号化パラメータ）が復号され、符号化装置106に供給されている。また、この例においては、磁気テープ604にユーザデータuser_dataとし

て、ヒストリ情報を記録する必要がないので、図 15 に示したヒストリエンコーディング装置 107 のうち、ヒストリ VLC211 のみが採用されている。そしてこのヒストリ VLC211 に、復号装置 102 が出力した符号化パラメータ（この例の場合、第 3 世代の符号化パラメータ）と、ヒストリデコーディング装置 104 がユーザデータ user_data から復号、出力した符号化パラメータ（この例の場合、第 1 世代と第 2 世代の符号化パラメータ）が供給されている。ヒストリ VLC211 は、この第 1 世代乃至第 3 世代の符号化パラメータを可変長符号化し、図 40 乃至図 46、または図 47 に示した history_stream を生成し、マルチプレクサ 621 に供給する。

【0632】マルチプレクサ 621 には、また、符号化装置 106 より出力された第 4 世代の符号化ストリーム ST が入力されている。マルチプレクサ 621 は、符号化装置 106 より供給された符号化ストリーム（ビットストリーム）をヒストリ VLC211 より供給されたヒストリよりも安全な領域に多重化する。

【0633】例えば、図 82 に示すように、磁気テープ 604 において、符号化装置 106 より出力されたビデオストリームは、シンクコードに近い位置に記録され、ヒストリ VLC211 より出力された history_stream は、シンクコードからビデオストリームより、より離れた位置に記録される。特殊再生時などにおいて、ビデオストリームを検索するとき、最初にシンクコードが検出され、そのシンクコードを基準として、それに続くビデオストリームが検索される。従って、シンクコードに近い位置にビデオストリームを配置した方が、高速再生時などにおいても、より確実にビデオデータを再生することが可能になる。history_stream は、高速再生時などにおいて、必要とされる情報ではない。そこで、この history_stream は、シンクコードからより離れた位置に配置しても、それほど支障は生じない。

【0634】マルチプレクサ 621 により多重化された信号は、チャンネルエンコーディング装置 602 に入力され、チャンネルエンコードされた後、記録ヘッド 603 により、磁気テープ 604 に記録される。

【0635】このように、この例においては、history_stream が、ビデオデータとは異なる位置に多重化されるため、仮にそこにスタートコードが現れたとしても、ビデオデータとは充分区別することが可能である。そこで、この例においては、マーカビットを挿入し、history_stream を converted_history_stream とする必要はない。

【0636】また、符号化パラメータを history_stream のフォーマットにしないで、そのままマルチプレクサ 621 に供給し、多重化させることも可能であるが、そのようにすると、圧縮されていないので、符号化パラメータのデータ量が多くなり、磁気テープ 604 の利用効率が低下する。そこで、ヒストリ VLC211 により圧

縮し、history_stream のフォーマットにして、多重化するようにするのが好ましい。

【0637】図 81 に示すように、ビデオテープレコーダ 601 の再生系においては、磁気テープ 604 から、再生ヘッド 611 により再生された信号が、チャンネルデコーディング装置 612 でチャンネルデコードされる。デマルチプレクサ 631 は、チャンネルデコーディング装置 612 でチャンネルデコードされる。デマルチプレクサ 631 は、チャンネルデコーディング装置 612 から供給された第 4 世代の符号化ストリーム ST を、ビデオストリームと、history_stream とに分離し、ビデオストリームを復号装置 102 に供給し、history_stream をヒストリ VLC203 に供給する。

【0638】すなわちこの例においては、図 15 に示したヒストリデコーディング装置 104 のうち、ヒストリ VLC203 のみが採用される。

【0639】ヒストリ VLC203 は、history_stream を可変長復号処理し、得られた第 1 世代乃至第 3 世代の符号化パラメータを符号化装置 106 に出力する。

【0640】また、デマルチプレクサ 631 より出力された history_stream は、コンバータ 212' に入力される。このコンバータ 212' と、後段のユーザデータフォーマッタ 213' は、ヒストリエンコーディング装置 107 に内蔵されているコンバータ 212、およびユーザデータフォーマッタ 213（図 15 参照）とは、別個のものであるが、それらと同一の機能を果たすものである。

【0641】すなわちコンバータ 212' は、デマルチプレクサ 631 より入力された history_stream にマーカビットを付加して、converted_history_stream を生成し、ユーザデータフォーマッタ 213' に出力する。ユーザデータフォーマッタ 213' は、入力された converted_history_stream を user_data に変換し、符号化装置 106 に出力する。この user_data には、第 1 世代乃至第 3 世代の符号化パラメータが含まれていることになる。

【0642】復号装置 102 は、デマルチプレクサ 631 から入力されたビデオストリームを復号し、ベースバンドビデオ信号を符号化装置 106 に出力する。また、復号装置 102 は、第 4 世代の符号化パラメータを符号化装置 106 に供給するとともに、ヒストリ VLC203 に出力する。ヒストリ VLC203 は、入力された第 4 世代の符号化パラメータから user_data を生成し、符号化装置 106 に出力する。

【0643】符号化装置 106 は、図 79 における符号化装置 106 と同様に、「通常符号化処理」または「パラメータ再利用符号化処理」を実行し、第 5 世代の符号化ストリーム ST を出力する。この第 5 世代の符号化ストリーム ST には、そのピクチャ層の user_data に、第 1 世

代乃至第4世代の符号化パラメータが記録されている。

【0644】本発明のトランスコードによれば、過去の符号化処理において生成された符号化パラメータを、現在の符号化処理において生成された符号化ストリームのユーザデータエリアに記述するようにし、生成されたビットストリームは、MPEG規格に準じた符号化ストリームであるので、既存のどのデコーダでも復号処理を行うことができる。さらには、本発明のトランスコードによれば、過去の符号化処理における符号化パラメータを伝送するために専用線のようなものを設ける必要がないので、従来のデータストリーム伝送環境をそのまま使用して、過去の符号化パラメータを伝送することができる。

【0645】本発明のトランスコードによれば、過去の符号化処理において生成された符号化パラメータを、選択的に現在の符号化処理において生成された符号化ストリーム中に記述するようにしているので、出力されるビットストリームのビットレートを極端に上げることなく、過去の符号化パラメータを伝送することができる。

【0646】本発明のトランスコードによれば、過去の符号化パラメータと現在の符号化パラメータの中から、現在の符号化処理に最適な符号化パラメータを選択して符号化処理を行うようにしているので、復号処理及び符号化処理を繰り返したとしても、画質劣化が蓄積されることはない。

【0647】本発明のトランスコードによれば、過去の符号化パラメータの中から、ピクチャタイプに応じて現在の符号化処理に最適な符号化パラメータを選択して符号化処理を行うようにしているので、復号処理及び符号化処理を繰り返したとしても、画質劣化が蓄積されることはない。

【0648】本発明のトランスコードによれば、過去の符号化パラメータに含まれるピクチャタイプに基づいて、過去の符号化パラメータを再利用するか否かを決定しているので、最適な符号化処理を行うことができる。

【0649】なお、上記各処理を行うコンピュータプログラムは、磁気ディスク、光ディスク、光磁気ディスク、半導体メモリなどの記録媒体に記録して提供するほか、インターネット、デジタル衛星などのネットワークを介して伝送し、ユーザの記録媒体に記録させることで提供することができる。

【0650】

【発明の効果】以上のごとく、請求項1に記載の符号化ストリーム記録装置、請求項5に記載の符号化ストリーム記録方法、および請求項6に記載の記録媒体に記録されているプログラムによれば、検出された過去の符号化処理における符号化履歴を記録媒体に記録するようにしたので、符号化ストリームを記録媒体に記録した場合においても、画質の劣化を抑制することが可能となる。

【0651】請求項7に記載の符号化ストリーム再生装置、請求項11に記載の符号化ストリーム再生方法、お

よび請求項12に記載の記録媒体のプログラムによれば、記録媒体から再生された符号化ストリームに含まれる符号化履歴を検出し、第2の符号化ストリームと多重化して出力するようにしたので、記録媒体から再生された符号化ストリームを再度トランスコーディングする場合においても、画質の劣化を抑制することが可能となる。

【0652】さらに、請求項13乃至請求項129に記載の発明においても、同様に、トランスコーディング時の画質の劣化を抑制することができる。

【図面の簡単な説明】

【図1】高効率符号化の原理を説明する図である。

【図2】画像データを圧縮する場合におけるピクチャタイプを説明する図である。

【図3】画像データを圧縮する場合におけるピクチャタイプを説明する図である。

【図4】動画像信号を符号化する原理を説明する図である。

【図5】動画像信号を符号化し、復号する装置の構成を示すブロック図である。

【図6】画像データの構成を説明する図である。

【図7】図5のエンコーダ18の構成を示すブロック図である。

【図8】図7の予測モード切替回路52の動作を説明する図である。

【図9】図7の予測モード切替回路52の動作を説明する図である。

【図10】図7の予測モード切替回路52の動作を説明する図である。

【図11】図7の予測モード切替回路52の動作を説明する図である。

【図12】図5のデコーダ31の構成を示すブロック図である。

【図13】ピクチャタイプに対応したSNR制御を説明する図である。

【図14】本発明を適用したトランスコード101の構成を示すブロック図である。

【図15】図14のトランスコード101のより詳細な構成を示すブロック図である。

【図16】図14の復号装置102に内蔵されるデコーダ111の構成を示すブロック図である。

【図17】マクロブロックの画素を説明する図である。

【図18】符号化パラメータが記録される領域を説明する図である。

【図19】図14の符号化装置106に内蔵されるエンコーダ121の構成を示すブロック図である。

【図20】図15のヒストリVLC211の構成例を示すブロック図である。

【図21】図15のヒストリVLD203の構成例を示すブロック図である。

【図 22】図 15 のコンバータ 212 の構成例を示すブロック図である。

【図 23】図 22 のスタッフ回路 323 の構成例を示すブロック図である。

【図 24】図 22 のコンバータ 212 の動作を説明するタイミングチャートである。

【図 25】図 15 のコンバータ 202 の構成例を示すブロック図である。

【図 26】図 25 のディリート回路 343 の構成例を示すブロック図である。

【図 27】図 15 のコンバータ 212 の他の構成例を示すブロック図である。

【図 28】図 15 のコンバータ 202 の他の構成例を示すブロック図である。

【図 29】図 15 のユーザデータフォーマット 213 の構成例を示すブロック図である。

【図 30】図 14 のトランスコード 101 が実際に使用される状態を示す図である。

【図 31】符号化パラメータが記録される領域を説明する図である。

【図 32】図 14 の符号化装置 106 の変更可能ピクチャタイプ判定処理を説明するフローチャートである。

【図 33】ピクチャタイプが変更される例を示す図である。

【図 34】ピクチャタイプが変更される他の例を示す図である。

【図 35】図 14 の符号化装置 106 の量子化制御処理を説明する図である。

【図 36】図 14 の符号化装置 106 の量子化制御処理を説明するフローチャートである。

【図 37】密結合されたトランスコード 101 の構成を示すブロック図である。

【図 38】ビデオシーケンスのストリームのシンタックスを説明する図である。

【図 39】図 38 のシンタックスの構成を説明する図である。

【図 40】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 41】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 42】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 43】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 44】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 45】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 46】固定長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 47】可変長の履歴情報を記録する history_stream() のシンタックスを説明する図である。

【図 48】sequence_header() のシンタックスを説明する図である。

【図 49】sequence_extension() のシンタックスを説明する図である。

【図 50】extension_and_user_data() のシンタックスを説明する図である。

【図 51】user_data() のシンタックスを説明する図である。

【図 52】group_of_pictures_header() のシンタックスを説明する図である。

【図 53】picture_header() のシンタックスを説明する図である。

【図 54】picture_coding_extension() のシンタックスを説明する図である。

【図 55】extension_data() のシンタックスを説明する図である。

【図 56】quant_matrix_extension() のシンタックスを説明する図である。

【図 57】copyright_extension() のシンタックスを説明する図である。

【図 58】picture_display_extension() のシンタックスを説明する図である。

【図 59】picture_data() のシンタックスを説明する図である。

【図 60】slice() のシンタックスを説明する図である。

【図 61】macroblock() のシンタックスを説明する図である。

【図 62】macroblock_modes() のシンタックスを説明する図である。

【図 63】motion_vectors(s) のシンタックスを説明する図である。

【図 64】motion_vector(r,s) のシンタックスを説明する図である。

【図 65】I ピクチャに対する macroblock_type の可変長符号を説明する図である。

【図 66】P ピクチャに対する macroblock_type の可変長符号を説明する図である。

【図 67】B ピクチャに対する macroblock_type の可変長符号を説明する図である。

【図 68】本発明を適用したトランスコード 101 の他の構成を示すブロック図である。

【図 69】履歴情報の項目の組み合わせを説明する図である。

【図 70】図 68 のトランスコード 101 の動作を説明するフローチャートである。

【図 71】re_coding_stream_info() のシンタックスを説明する図である。

【図 7 2】 red_bw_flag, red_bw_indicator を説明する図である。

【図 7 3】 Re_Coding Information Bus macroblock for mation を説明する図である。

【図 7 4】 Picture rate elements を説明する図である。

【図 7 5】 Picture rate elements を説明する図である。

【図 7 6】 Picture rate elements を説明する図である。

【図 7 7】 Re_Coding Information Bus が記録される領域を説明する図である。

【図 7 8】 ビデオテープレコーダの記録系の構成例を表すブロック図である。

【図 7 9】 ビデオテープレコーダの再生系の構成例を表すブロック図である。

【図 8 0】 ビデオテープレコーダの記録系の他の構成例を表すブロック図である。

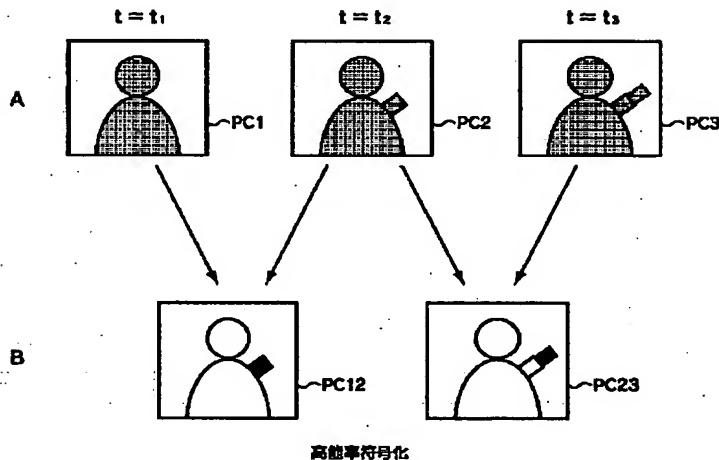
【図 8 1】 ビデオテープレコーダの再生系の他の構成例を表すブロック図である。

【図 8 2】 ビデオストリームと history_stream の記録位置を説明する図である。

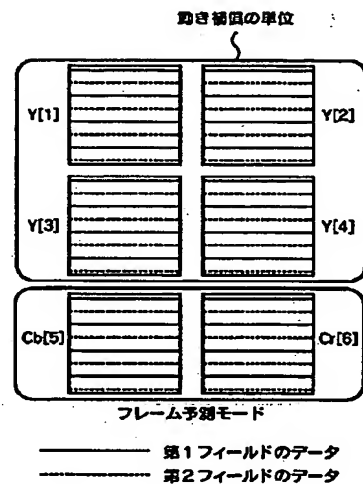
【符号の説明】

1 符号化装置, 2 復号装置, 3 記録媒体,
12, 13 A/D変換器, 14 フレームメモリ,
15 輝度信号フレームメモリ, 16 色差信号フレームメモリ, 17 フォーマット変換回路, 18 エンコーダ, 31 デコーダ, 32 フォーマット変換回路, 33 フレームメモリ, 34 輝度信号フレームメモリ, 35 色差信号フレームメモリ, 36, 37 D/A変換器, 50 動きベクトル検出回路, 51 フレームメモリ, 52 予測モード切り替え回路, 53 演算器, 54 予測判定回路, 55 DCTモード切り替え回路, 56 DCT回路, 57 量子化回路, 58 可変長符号化回路, 59 送信バッファ, 60 逆量子化回路, 61 IDCT回路, 62 演算器, 63 フレームメモリ, 64 動き補償回路, 81 受信バッファ, 82 可変長復号回路, 83 逆量子化回路, 84 IDCT回路, 85 演算器, 86 フレームメモリ, 87 動き補償回路, 101 トランスコーダ, 102 復号装置, 103 ヒストリ情報多重化装置, 105 ヒストリ情報分離装置, 106 符号化装置, 111 デコーダ, 112 可変長復号回路, 121 エンコーダ,

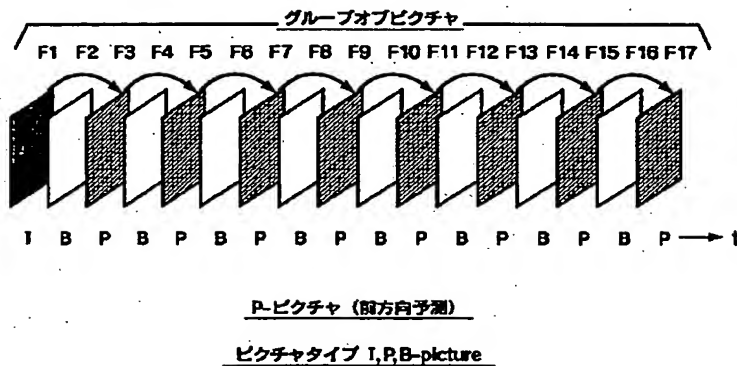
【図 1】



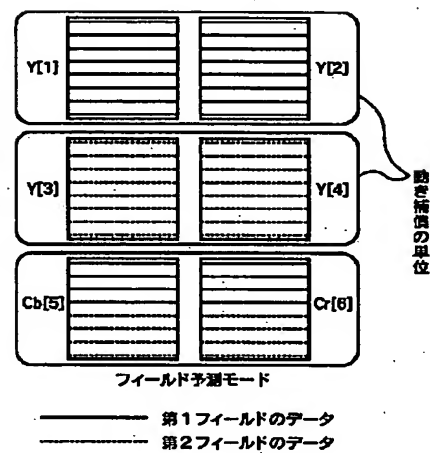
【図 8】



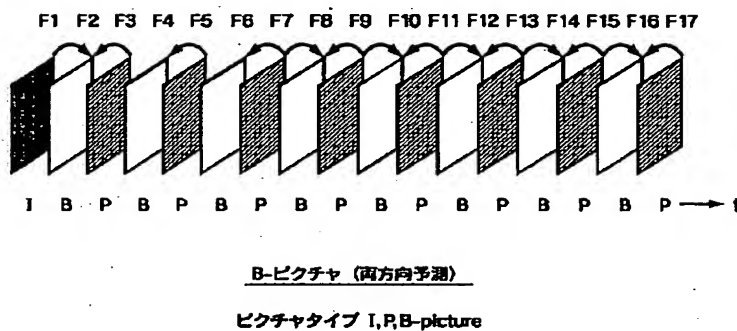
【図 2】



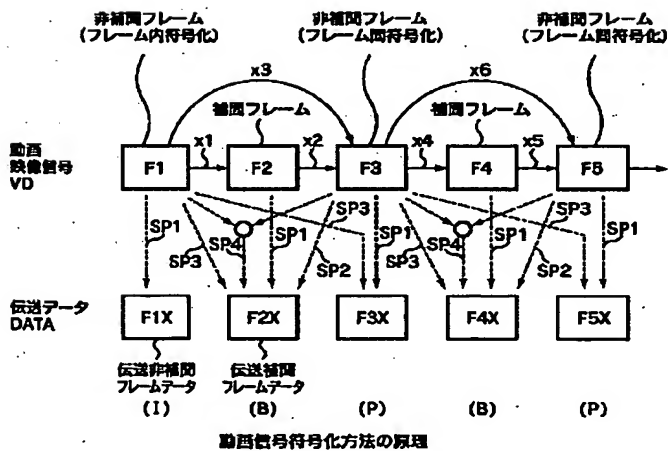
【図9】



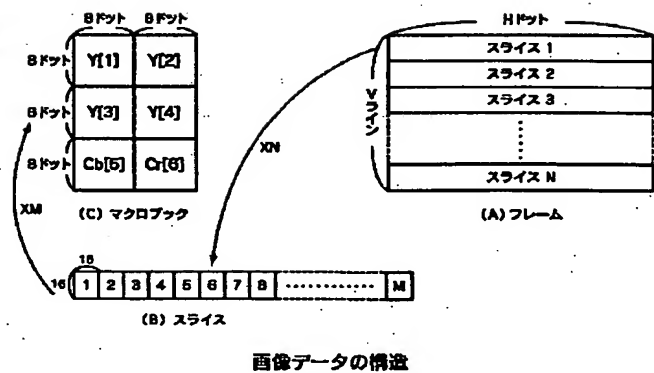
【図 3】



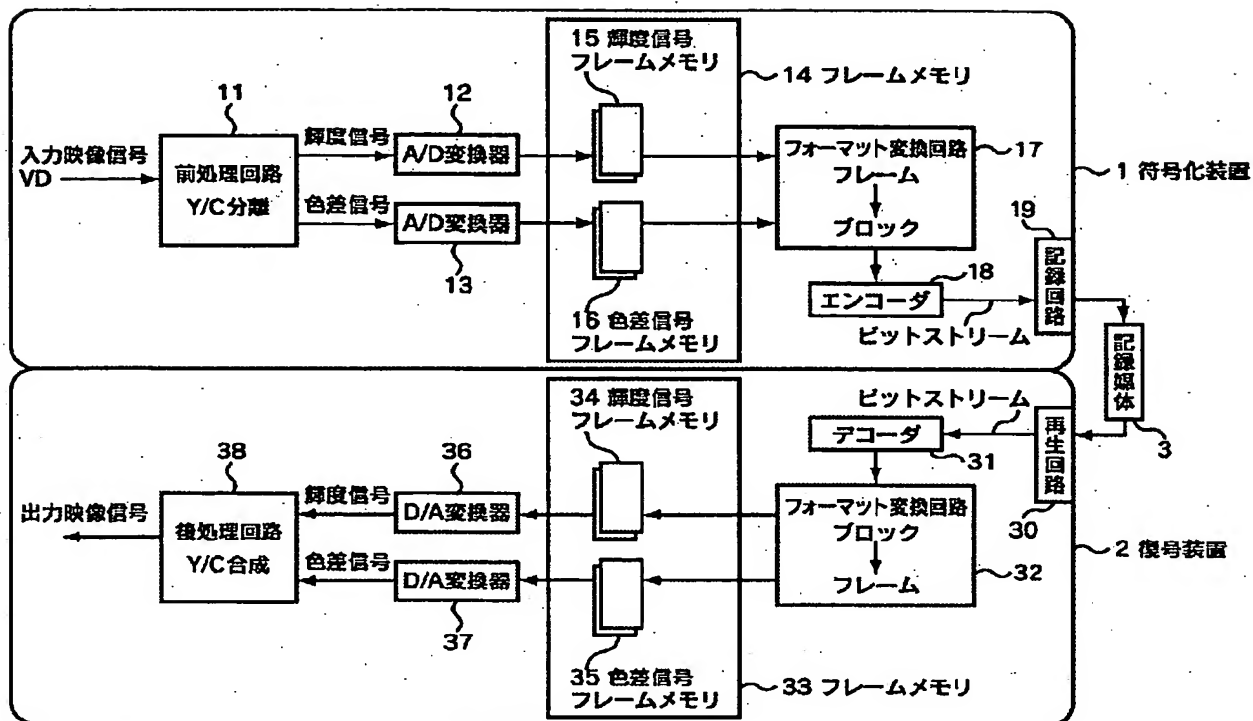
【图 4】



【図 6】

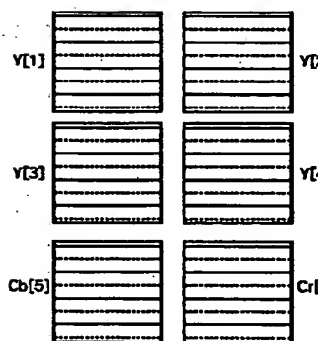


【図5】



動画像符号化／復号装置

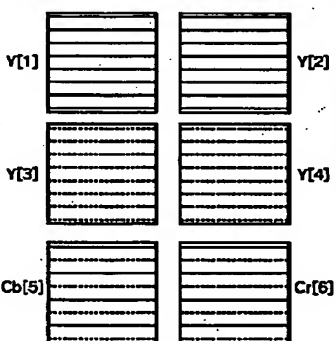
【図10】



フレームDCTモード

—— 第1フィールドのデータ
 - - - 第2フィールドのデータ

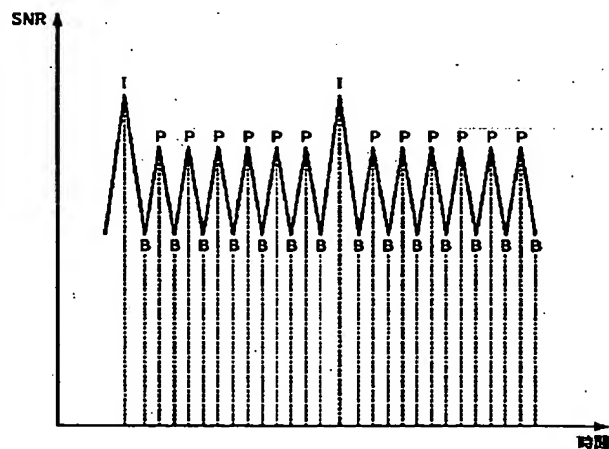
【図11】



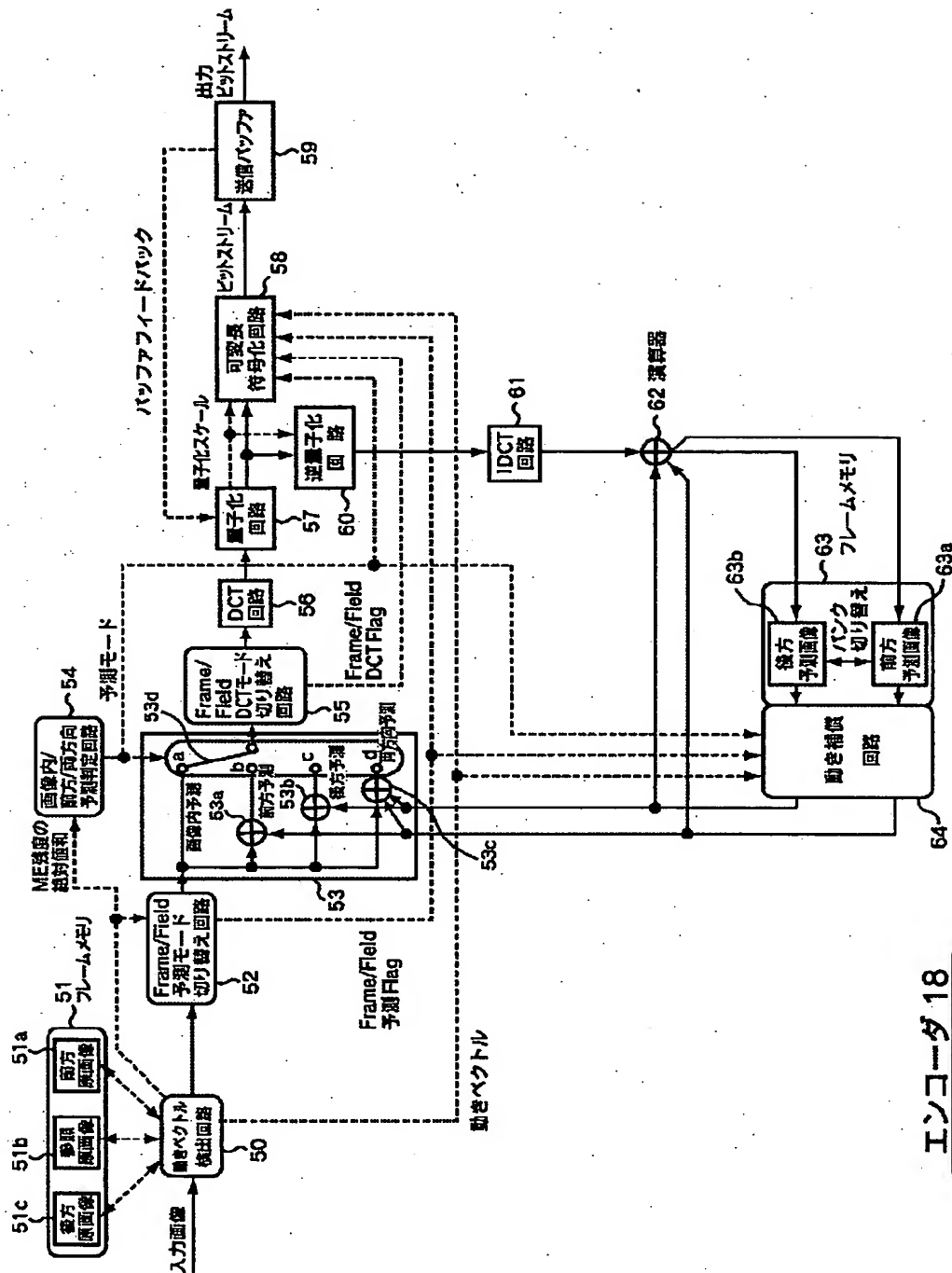
フィールドDCTモード

—— 第1フィールドのデータ
 - - - 第2フィールドのデータ

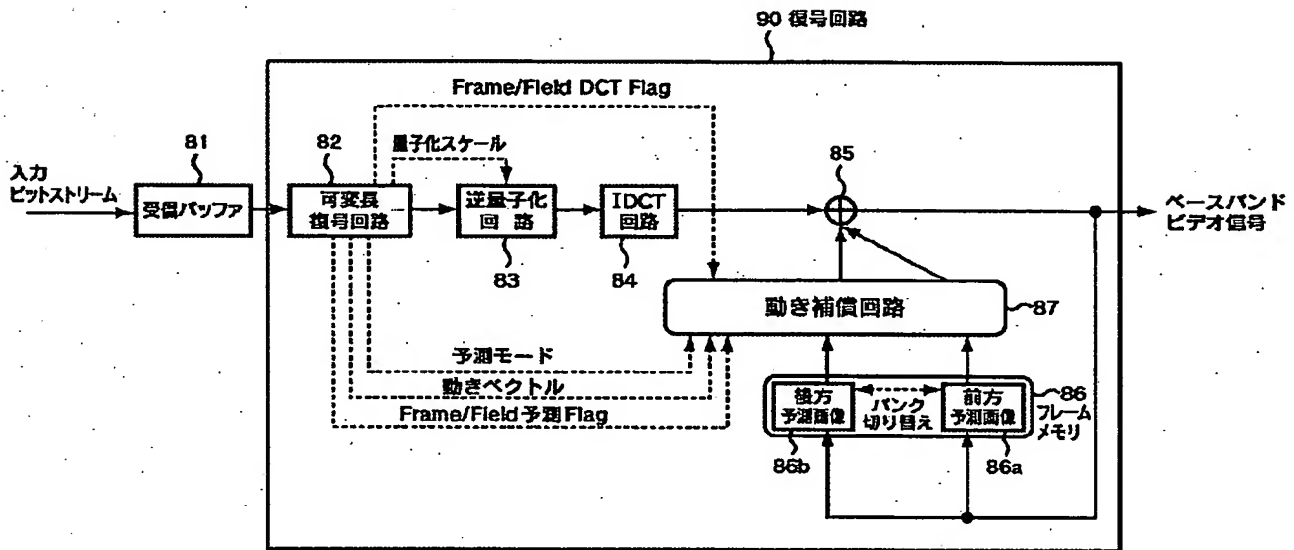
【図13】



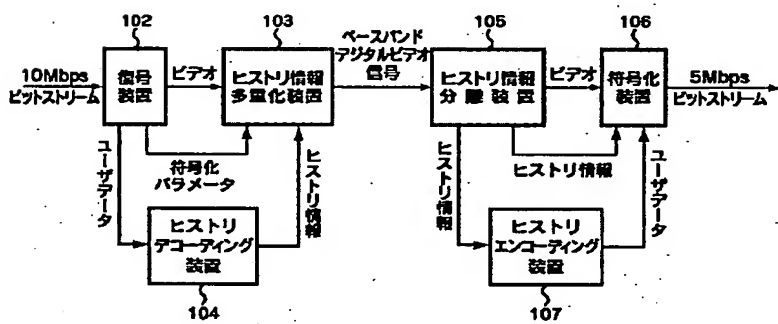
【図7】



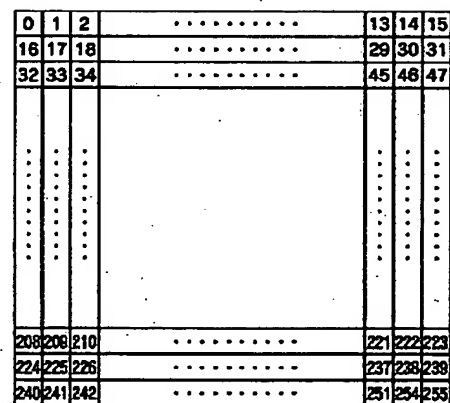
デコーダ 31



【图 17】

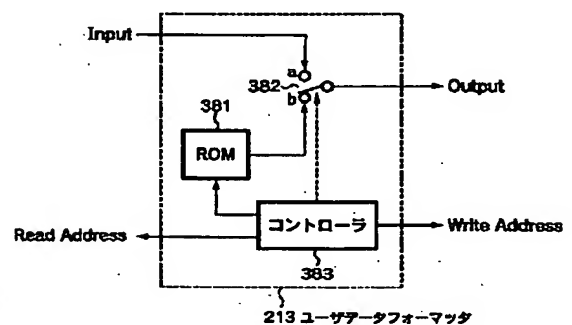
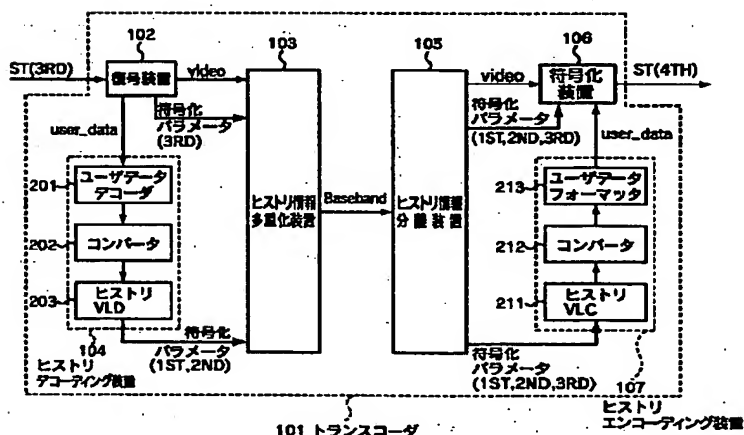


トランスコーダ 101

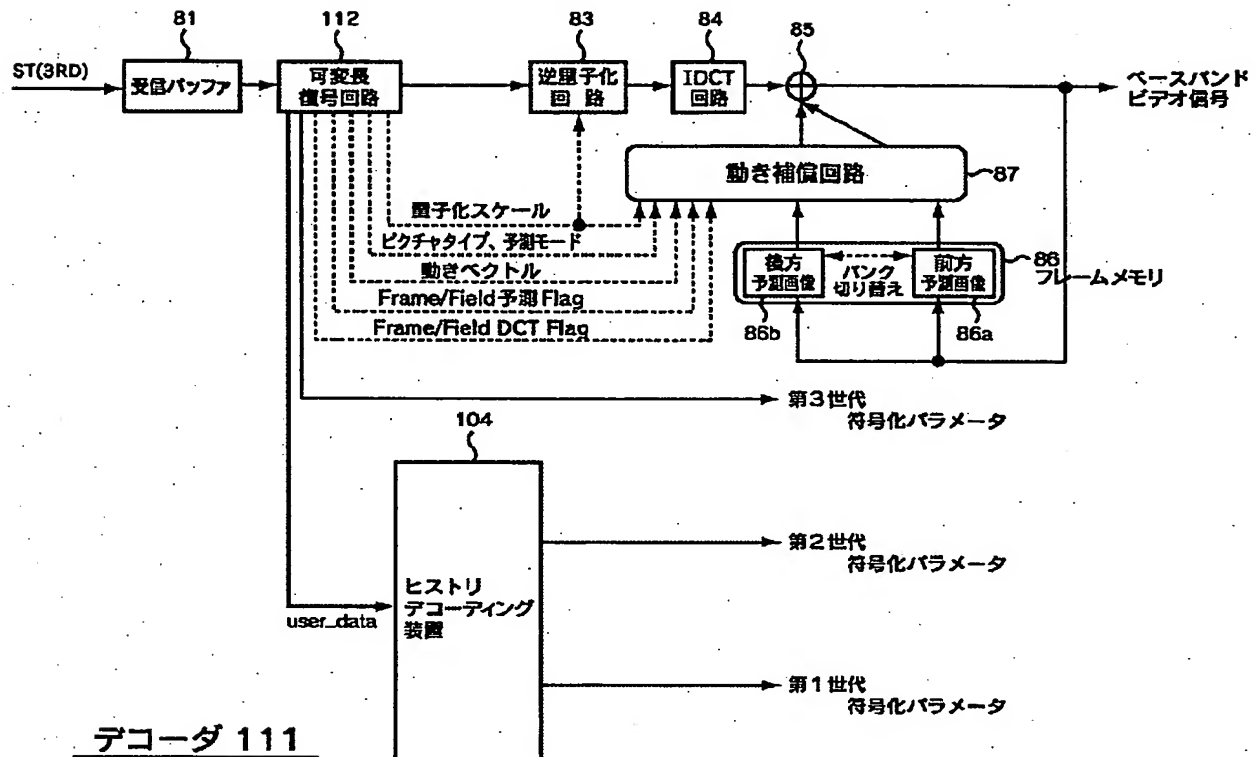


マクロブロック

【图 29】



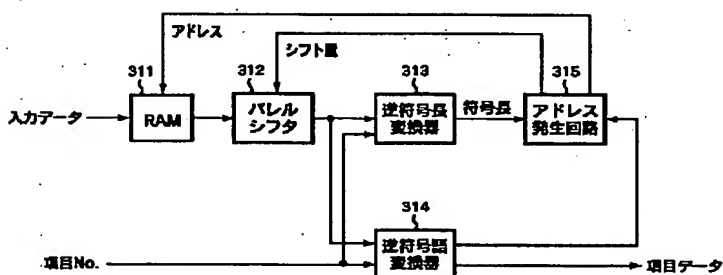
【図 16】



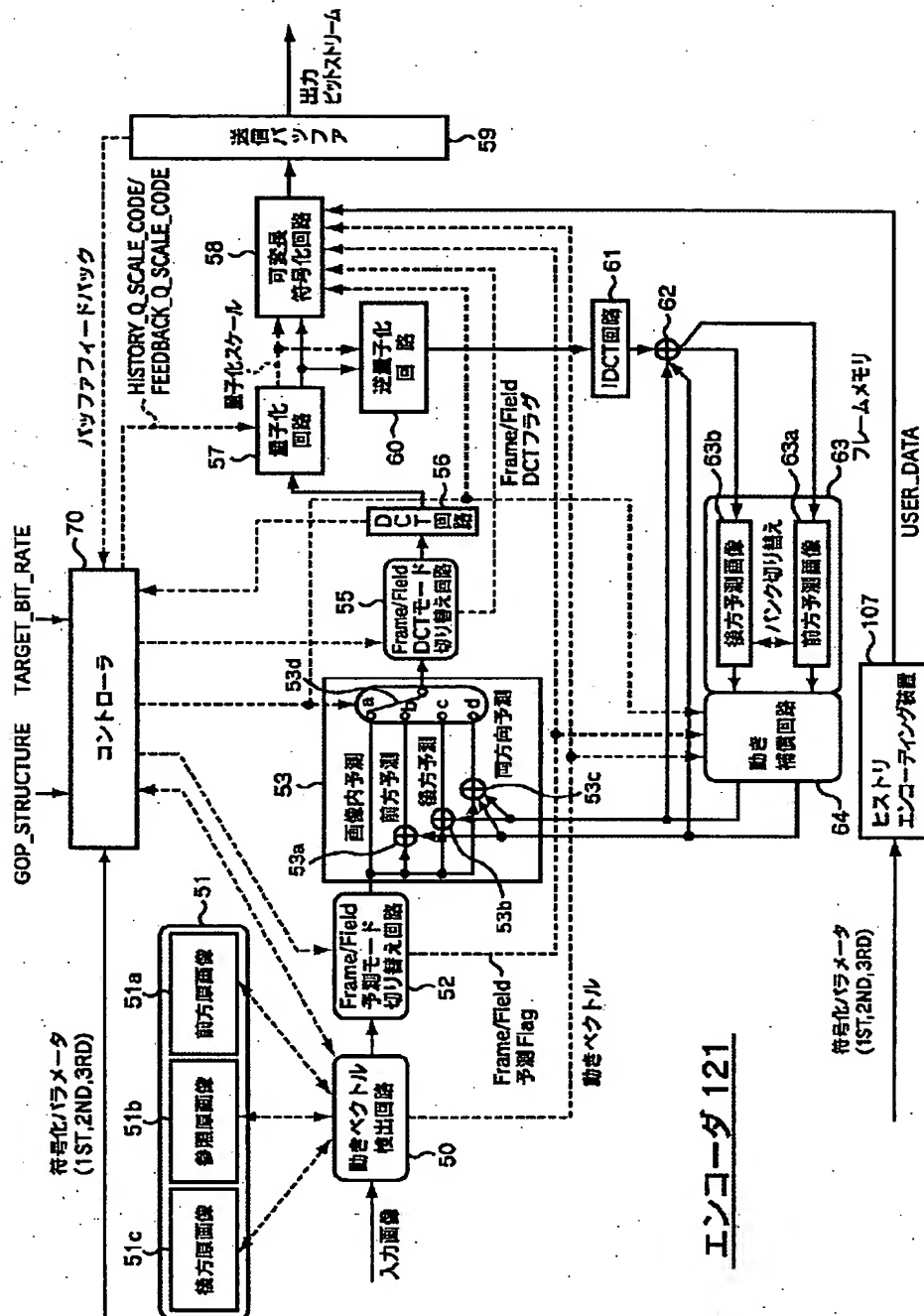
【図 18】

D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]	画像データ領域
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]	
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]	
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]	
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]	
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]	
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]	
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]	
D1	第1世代		第2世代		第3世代				ヒストリ情報領域
D0									
	Cb[0][0]	Y[0][0]	Cr[0][0]	Y[1][0]	Cb[1][0]	Y[2][0]	Cr[1][0]	Y[3][0]	

【図 21】

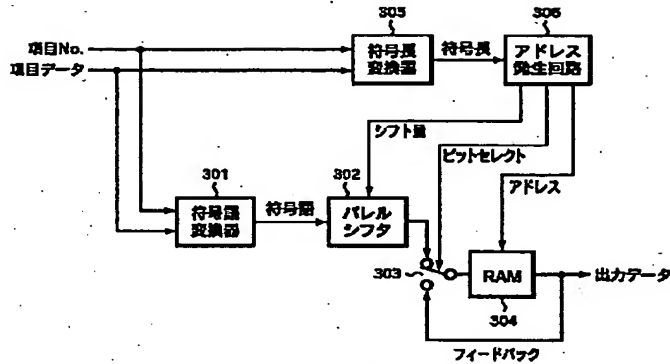


【図 19】



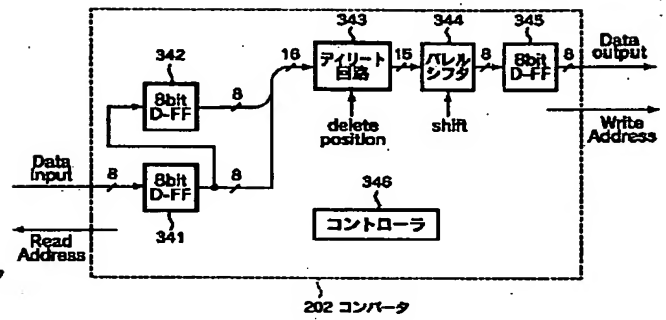
エンコーダ 121

【図 20】

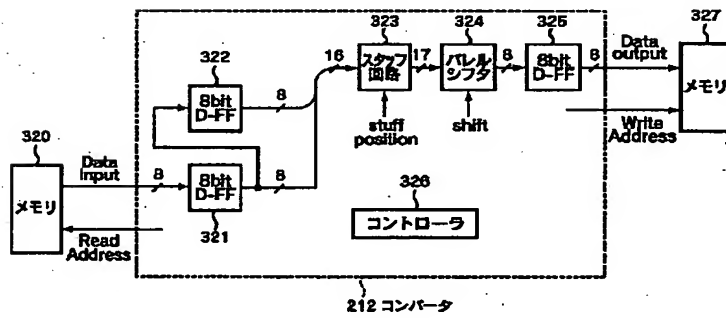


ヒストリフォーマッタ 211

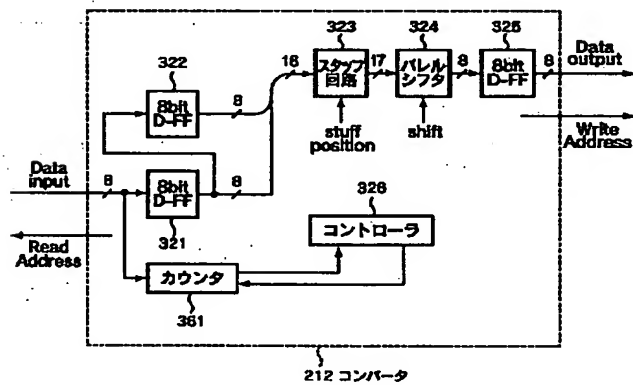
【図 25】



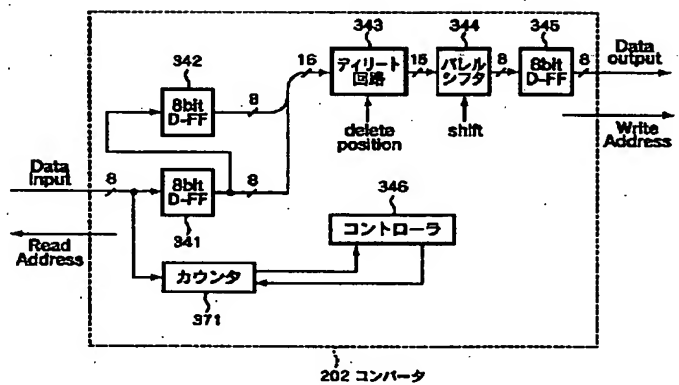
【図 22】



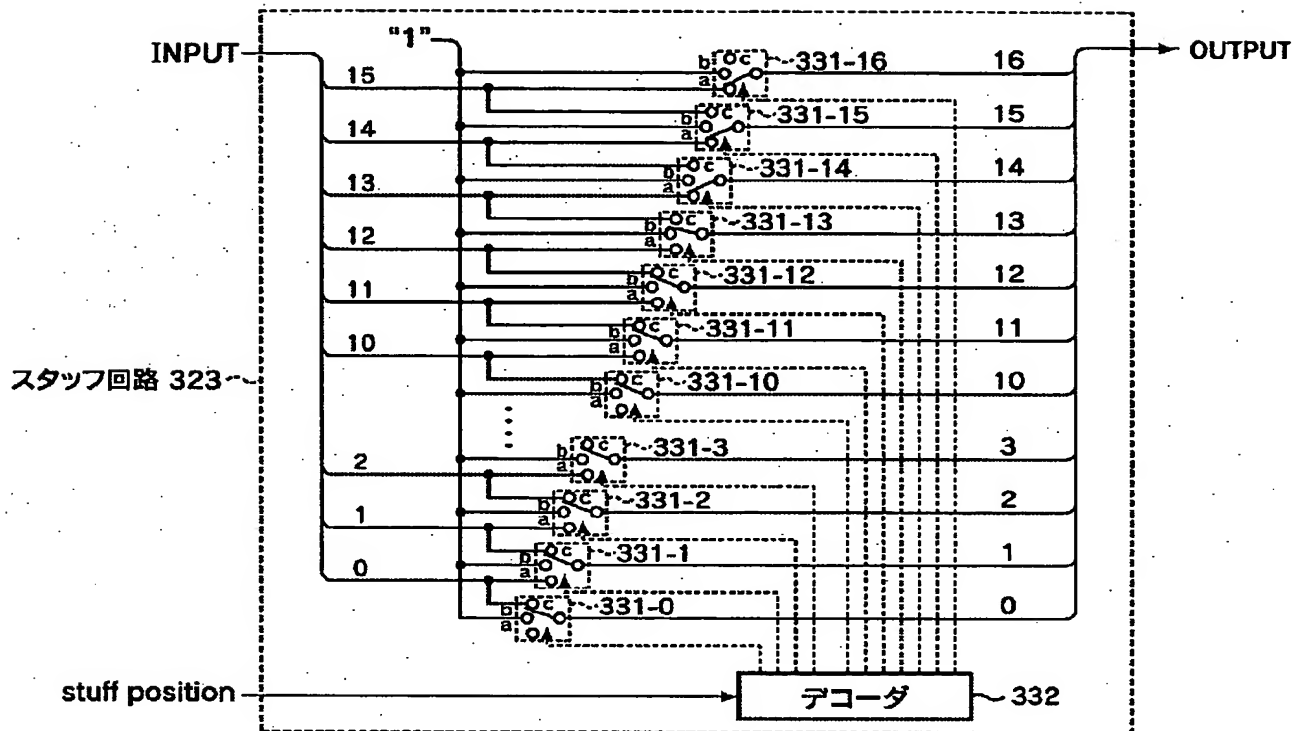
【図 27】



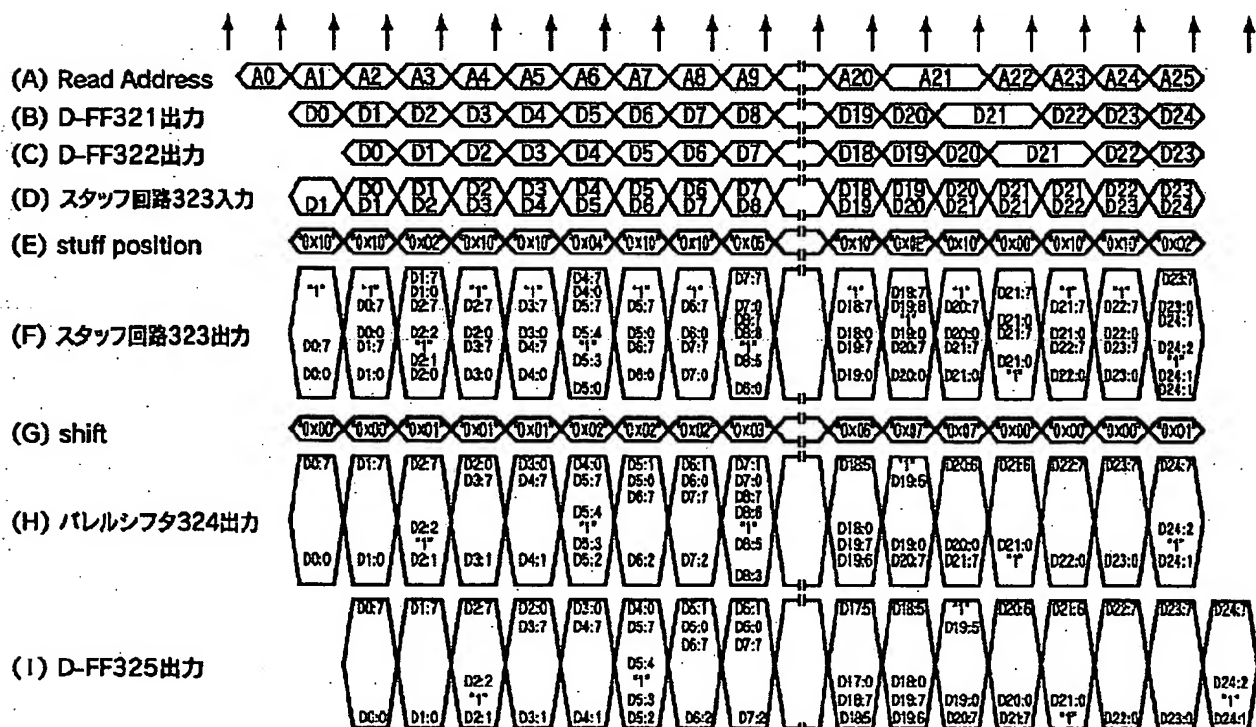
【図 28】



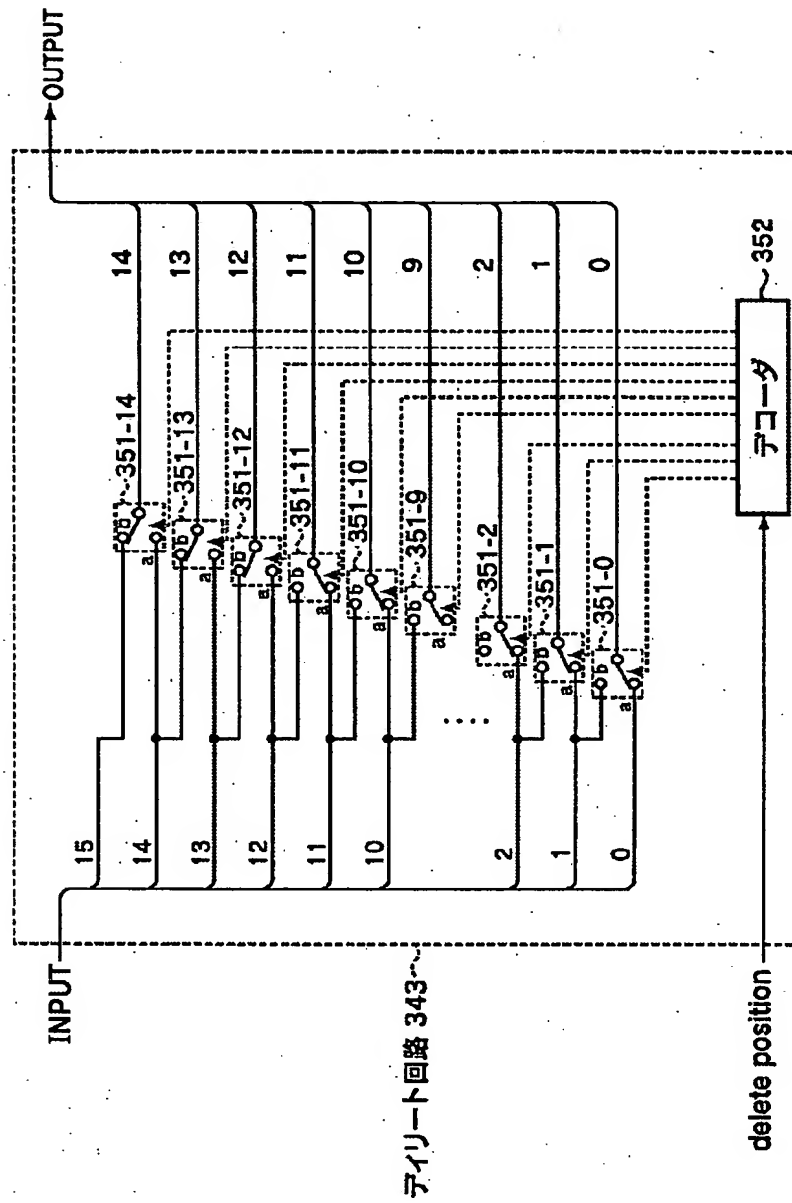
【図 2 3】



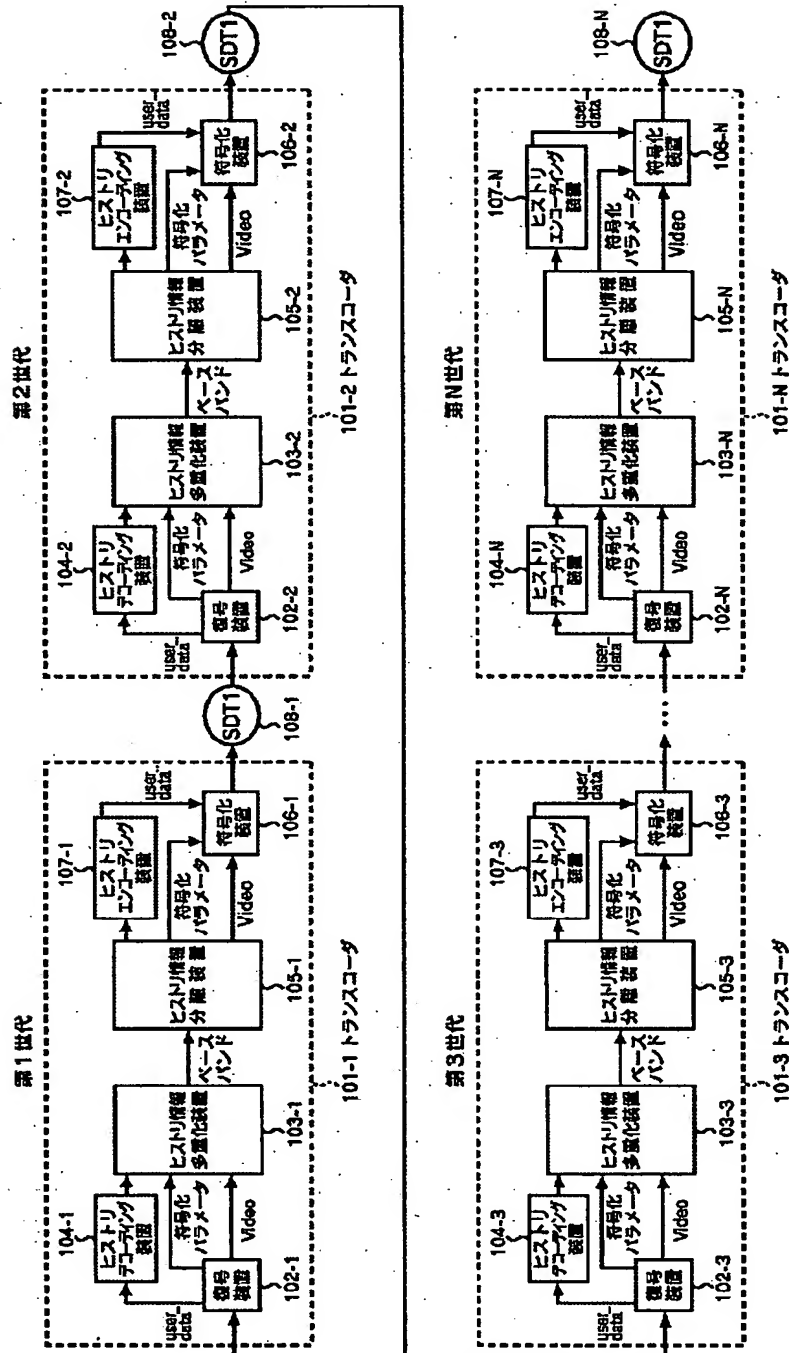
【図 2 4】



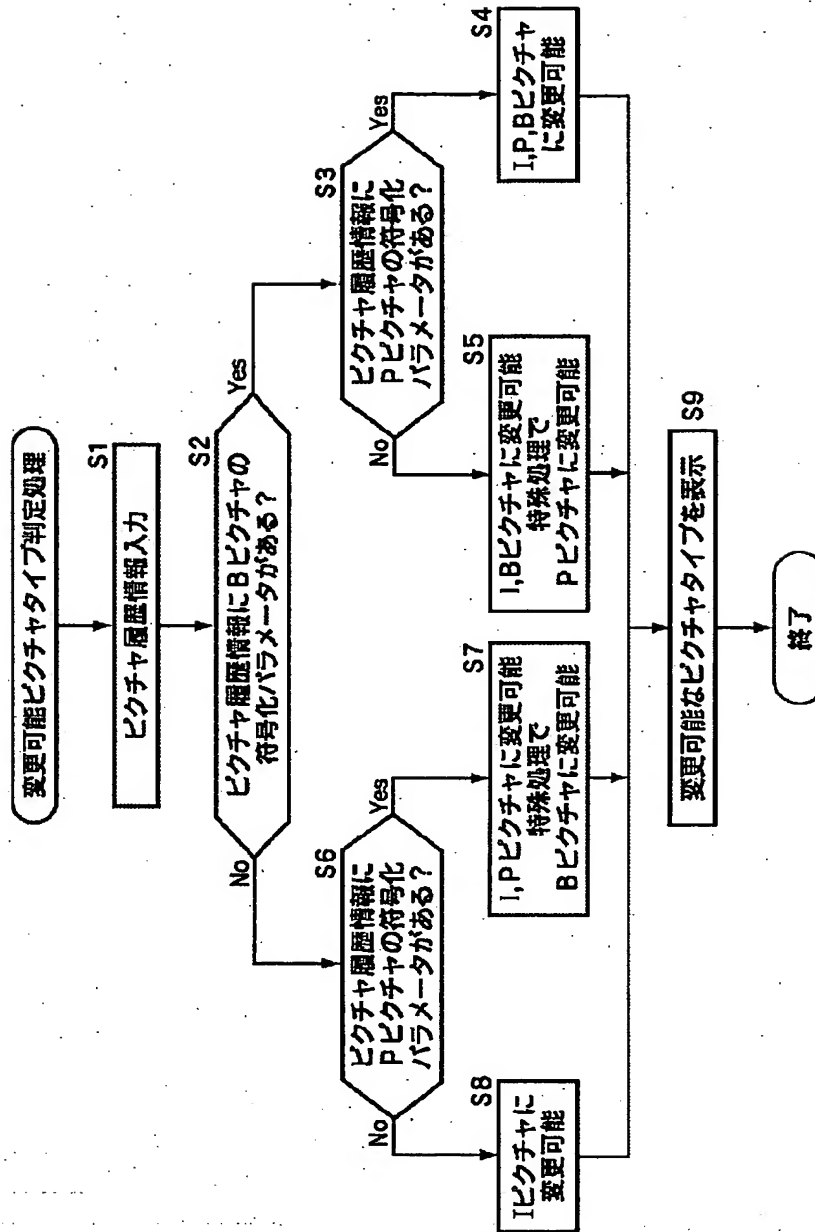
【図26】



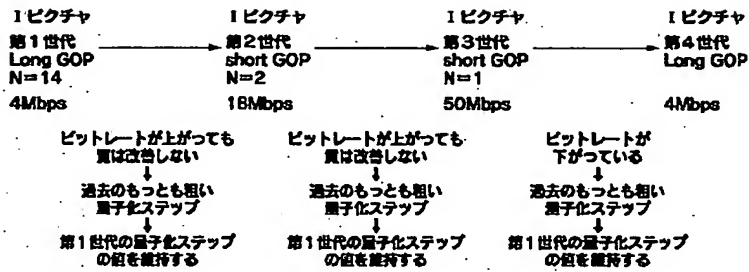
【図30】



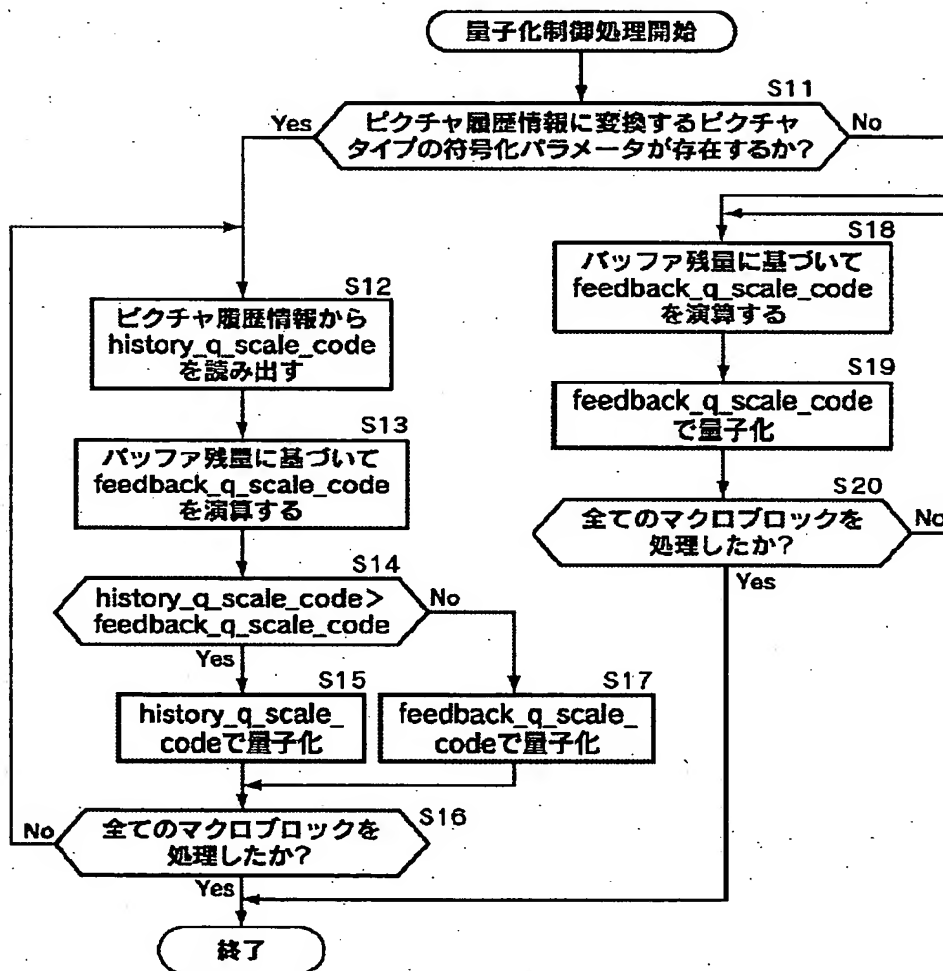
【図 32】



【図 35】



【図 36】



【図 46】

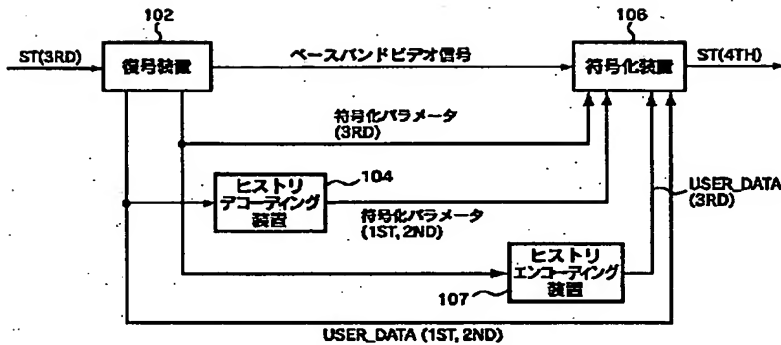
History stream(40-7)

	bits	value
num_other_bits	7	
marker_bit	1	1

【図 5.1】

user_data () {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
while(nextbits() != '0000 0000 0000 0000 0001') {		
user_data	8	ulmsbf
}		
next_start_code()		
}		

【図 37】



【図 38】

stream with history data		
	No. of bits	Mnemonic
video_sequence () {		
next_start_code ()		
sequence_header ()		
sequence_extension ()		
do {		
extension_and_user_data (0)		
do {		
if (nextbits () == group_start_code) {		
group_of_pictures_header (1)		
extension_and_user_data (1)		
}		
picture_header ()		
picture_coding_extension ()		
while (nextbits () == extension_start_code)		
(nextbits () == user_data_start_code) {		
if (nextbits () == extension_start_code) {		
extension_data (2)		
}		
if (nextbits () == user_data_start_code) {		
user_data_start_code	32	bslbf
}		
if (nextbits () == History_Data_ID) {		
History_Data_ID	8	bslbf
converted_history_stream ()		
}		
else {		
user_data ()		
}		
}		
picture_data ()		
while (nextbits () == picture_start_code)		
(nextbits () == group_start_code) {		
if (nextbits () == sequence_end_code) {		
sequence_header ()		
sequence_extension ()		
}		
}		
while (nextbits () != sequence_end_code)		
sequence_end_code	32	bslbf
}		

【図 50】

extension_and_user_data (1) {		
	No. of bits	Mnemonic
while (nextbits () == extension_start_code)		
(nextbits () == user_data_start_code) {		
if ((0 == 2) && (nextbits () == extension_start_code)) {		
extension_data ()		
}		
if (nextbits () == user_data_start_code) {		
user_data ()		
}		
}		
}		

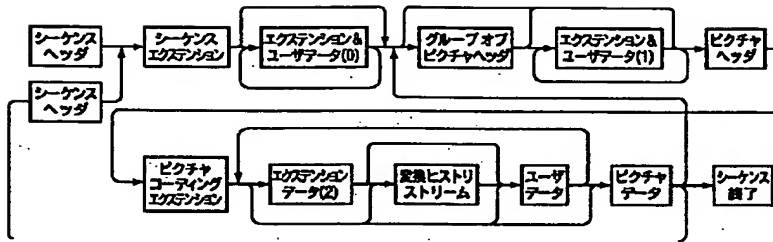
【図 40】

history stream (40-1)		
	bits	value
sequence_header		
sequence_header_code	32	000001B3
sequence_header_present_flag	1	
horizontal_size_value	12	
marker_bit	1	1
vertical_size_value	12	
aspect_ratio_information	4	
frame_rate_code	4	
marker_bit	1	1
bit_rate_value	18	
marker_bit	1	1
vbv_buffer_size_value	10	
constrained_parameter_flag	1	0
load_intra_quantiser_matrix	1	
load_non_intra_quantiser_matrix	1	
marker_bits	5	1F
intra_quantiser_matrix[64]	8*64	
non_intra_quantiser_matrix[64]	8*64	
sequence_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	1
sequence_extension_present_flag	1	
profile_and_level_indication	8	
progressive_sequence	1	
chroma_format	2	
horizontal_size_extension	2	
vertical_size_extension	2	
marker_bit	1	1
bit_rate_extension	12	
vbv_buffer_size_extension	8	
low_delay	1	
marker_bit	1	1

【図 52】

group_of_pictures_header () {		
	No. of bits	Mnemonic
group_start_code	32	bslbf
time_code	25	bslbf
closed_gap	1	uimbsf
broken_link	1	uimbsf
next_start_code ()		
}		

【図 39】



【図 41】

history stream(40-2)

	bits	value
frame_rate_extension_n	2	
frame_rate_extension_d	5	
marker_bits	6	3F
sequence_display_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	2
sequence_display_extension_present_flag	1	
video_format	3	
colour_description	1	
colour_primaries	8	
transfer_characteristics	8	
marker_bit	1	1
matrix_coefficients	8	
display_horizontal_size	14	
marker_bit	1	1
display_vertical_size	14	
marker_bit	1	1
macroblock_assignment_in_user_data		
macroblock_assignment_present_flag	1	
marker_bit	7	7F
v_phase	8	
h_phase	8	
group_of_picture_header		
group_start_code	32	000001B8
group_of_picture_header_present_flag	1	
time_code	25	
closed_gop	1	
broken_link	1	
marker_bits	4	F
picture_header		
picture_start_code	32	00000100

【図 47】

history_stream()	No. of bits	Mnemonic
next_start_code()		
sequence_header()		
sequence_extension()		
extension_and_user_data(0)		
if(nextbits() == group_start_code){		
group_of_pictures_header()		
extension_and_user_data(1)		
}		
picture_header()		
picture_coding_extension()		
re_coding_stream_info()		
extensions_and_user_data(2)		
picture_data()		
sequence_end_code	32	bslbf

【図 42】

history stream(40-3)

	bits	value
temporal_reference	10	
picture_coding_type	3	
marker_bit	1	1
vbv_delay	16	
full_pel_forward_vector	1	
forward_f_code	3	
full_pel_backward_vector	1	
marker_bit	1	1
backward_f_code	3	
marker_bit	1	1
picture_coding_extension		
extension_start_code	32	000001B5
extension_start_code_identifier	4	8
f_code[0][0]	4	
f_code[0][1]	4	
f_code[1][0]	4	
f_code[1][1]	4	
intra_dc_precision	2	
picture_structure	2	
top_field_first	1	
frame_pred_frame_dct	1	
concealment_motion_vectors	1	
q_scale_type	1	
marker_bit	1	1
intra_vlc_format	1	
alternate_scan	1	
repeat_first_field	1	
chroma_420_type	1	
progressive_frame	1	
composite_display_flag	1	
v_axis	1	
field_sequence	3	
sub_carrier	1	
burst_amplitude	7	

【図 55】

extension_data()	No. of bits	Mnemonic
while(nextbits() == extension_start_code) {		
extension_start_code	32	bslbf
if (nextbits() == "Quant Matrix Extension ID")		
quant_matrix_extension()		
else if (nextbits() == "Copyright Extension ID")		
copyright_extension()		
else		
picture_display_extension()		
}		

【図 43】

history stream(40-4)		
	bits	value
marker_bits	1	1
sub_carrier_phase	8	
quant_matrix_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	3
quant_matrix_extension_present_flag	1	
load_intra_quantiser_matrix	1	
marker_bits	2	3
intra_quantiser_matrix[64]	8*64	
load_non_intra_quantiser_matrix	1	
marker_bits	7	7F
non_intra_quantiser_matrix[64]	8*64	
load_chroma_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_intra_quantiser_matrix[64]	8*64	
load_chroma_non_intra_quantiser_matrix	1	
marker_bits	7	7F
chroma_non_intra_quantiser_matrix[64]	8*64	
copyright_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	4
copyright_extension_present_flag	1	
copyright_flag	1	
copyright_identifier	8	
original_or_copy	1	
marker_bit	1	
copyright_number_1	20	
marker_bit	1	
copyright_number_2	22	
marker_bit	1	
copyright_number_3	22	3F
marker_bits	6	

【図 48】

sequence_header() {		
	No. of bits	Mnemonic
sequence_header_code()	32	bslbf
horizontal_size_value	12	ulmsbf
vertical_size_value	12	ulmsbf
aspect_ratio_information	4	ulmsbf
frame_rate_code	4	ulmsbf
bit_rate_value	16	ulmsbf
marker_bit	1	bslbf
vbv_buffer_size_value	10	ulmsbf
constrained_parameters_flag	1	bslbf
load_intra_quantiser_matrix	1	ulmsbf
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix[64]	8*64	ulmsbf
load_non_intra_quantiser_matrix	1	ulmsbf
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix[64]	8*64	ulmsbf
next_start_code()		
}		

【図 59】

picture_data() {		
	No. of bits	Mnemonic
while(nextbits()==slice_start_code) {		
slice()		
}		
next_start_code()		
}		

【図 44】

history stream(40-5)		
	bits	value
picture_display_extension		
extension_start_code	32	00000185
extension_start_code_identifier	4	7
picture_display_extension_present_flag	1	
frame_centre_horizontal_offset_1	16	
marker_bit	1	1
frame_centre_vertical_offset_1	16	
marker_bit	1	1
frame_centre_horizontal_offset_2	16	
marker_bit	1	1
frame_centre_vertical_offset_2	16	
marker_bit	1	1
frame_centre_horizontal_offset_3	16	
marker_bit	1	1
frame_centre_vertical_offset_3	16	
marker_bit	6	3F
re_coding_stream_information		
user_data_start_code	32	00000182
re_coding_stream_info_id	16	91EC
red_bw_flag	1	
red_bw_indicator	2	
marker_bit	5	1F
user_data		
user_data_start_code	32	00000182
user_data	2048	
while(macroblock != macroblock_count) {		
macroblock		
macroblock_address_h	8	
macroblock_address_v	8	
slice_header_present_flag	1	
skipped_macroblock_flag	1	
marker_bit	1	1
macroblock_modes()		
macroblock_quant	1	
macroblock_motion_forward	1	
macroblock_motion_backward	1	
macroblock_pattern	1	
macroblock_intra	1	

【図 49】

sequence_extension() {		
	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	ulmsbf
profile_and_level_indication	8	ulmsbf
progressive_sequence	1	ulmsbf
chroma_format	2	ulmsbf
horizontal_size_extension	2	ulmsbf
vertical_size_extension	2	ulmsbf
bit_rate_extension	12	ulmsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	ulmsbf
low_delay	1	ulmsbf
frame_rate_extension_n	2	ulmsbf
frame_rate_extension_d	5	ulmsbf
next_start_code()		
}		

【図 58】

picture_display_extension() {		
	No. of bits	Mnemonic
extension_start_code_identifier	4	ulmsbf
for(i=0;i<number_of_frame_centre_offsets;i++) {		
frame_centre_horizontal_offset	16	slmsbf
marker_bit	1	bslbf
frame_centre_vertical_offset	16	slmsbf
marker_bit	1	bslbf
}		
next_start_code()		
}		

【図 45】

history stream(40-6)

	bits	value
spatial_temporal_weight_code_flag	1	
frame_motion_type	2	
field_motion_type	2	
dct_type	1	
marker_bits	2	3
quantiser_scale_code	5	
marker_bits	3	7
PMV[0][0][0]	14	
marker_bits	2	3
PMV[0][0][1]	14	
motion_vertical_field_select[0][0]	1	
marker_bit	1	1
PMV[0][1][0]	14	
marker_bits	2	3
PMV[0][1][1]	14	
motion_vertical_field_select[0][1]	1	
marker_bit	1	1
PMV[1][0][0]	14	
marker_bits	2	3
PMV[1][0][1]	14	
motion_vertical_field_select[1][0]	1	
marker_bit	1	1
PMV[1][1][0]	14	
marker_bits	2	3
PMV[1][1][1]	14	
motion_vertical_field_select[1][1]	1	
marker_bit	1	1
coded_block_pattern	12	
marker_bits	4	F
num_mv_bits	8	
num_coef_bits	14	
marker_bits	2	3

【図 54】

picture_coding_extension()	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code[0][0] /*forward horizontal*/	4	uimsbf
f_code[0][1] /*forward vertical*/	4	uimsbf
f_code[1][0] /*backward horizontal*/	4	uimsbf
f_code[1][1] /*backward vertical*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag){		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code()		

【図 53】

picture_header()	No. of bits	Mnemonic
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
forward_f_code	3	bslbf
}		
if(picture_coding_type==3){		
full_pel_forward_vector	1	bslbf
backward_f_code	3	bslbf
}		
while(nextbits()=='1'){		
extra_bit_picture/*with the value'1'*/	1	uimsbf
extra_information_picture	8	uimsbf
}		
extra_bit_picture/*with the value'0'*/	1	uimsbf
next_start_code()		

【図 56】

quant_matrix_extension()	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
load_intra_quantiser_matrix	1	uimsbf
if (load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	uimsbf
if (load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_intra_quantiser_matrix	1	uimsbf
if (load_chroma_intra_quantiser_matrix)		
chroma_intra_quantiser_matrix [64]	8*64	uimsbf
load_chroma_non_intra_quantiser_matrix	1	uimsbf
if (load_chroma_non_intra_quantiser_matrix)		
chroma_non_intra_quantiser_matrix [64]	8*64	uimsbf
next_start_code()		

【図 57】

copyright_extension()	No. of bits	Mnemonic
extension_start_code_identifier	4	uimsbf
copyright_flag	1	bslbf
copyright_identifier	8	uimsbf
original_or_copy	1	bslbf
reserved	7	uimsbf
marker_bit	1	bslbf
copyright_number_1	20	uimsbf
marker_bit	1	bslbf
copyright_number_2	22	uimsbf
marker_bit	1	bslbf
copyright_number_3	22	uimsbf
next_start_code()		

【図 82】

604

シンクコード	ビデオストリーム	history_stream
--------	----------	----------------

【図 60】

slice() {	No. of bits	Mnemonic
slice_start_code	32	bslbf
slice_quantiser_scale_code	5	uimsbf
if (nextbit()=='1') {		
intra_slice_flag	1	bslbf
intra_slice	1	uimsbf
reserved_bits	7	uimsbf
while(nextbits()=='1') {		
extra_bit_slice/*with the value '1'*/	1	uimsbf
extra_information_slice	8	uimsbf
}		
}		
extra_bit_slice/*with the value '0'*/	1	uimsbf
do {		
macroblock()		
} while(nextbit()!= '000 0000 0000 0000 0000')		
next_start_code()		
}		

【図 61】

macroblock() {	No. of bits	Mnemonic
while(nextbits()=='0000 0001 000')		
macroblock_escape	11	bslbf
macroblock_address_increment	1-11	viclbf
macroblock_modes()		
if (macroblock_quant)		
macroblock_quantiser_scale_code	5	uimsbf
if (macroblock_motion_forward		
(macroblock_intra && concealment_motion_vectors))		
motion_vectors(0)		
if (macroblock_motion_backward)		
motion_vectors(1)		
if (macroblock_intra && concealment_motion_vectors)		
marker_bit	1	bslbf
}		

【図 62】

macroblock_modes() {	No. of bits	Mnemonic
macroblock_type	1-9	viclbf
if (macroblock_motion_forward		
macroblock_motion_backward) {		
if (picture_structure=='frame') {		
if (frame_pred_frame_dct==0)		
frame_motion_type	2	uimsbf
} else {		
field_motion_type	2	uimsbf
}		
}		
if ((picture_structure=='Frame picture') &&		
(frame_pred_frame_dct==0) &&		
(dct_type_flag==1)) {		
dct_type	1	uimsbf
}		

【図 63】

motion_vectors(s) {	No. of bits	Mnemonic
if (motion_vector_count==1) {		
if ((mv_format==field)&&(dmv!=1))		
motion_vertical_field_select [0][s]	1	uimsbf
motion_vector(0,s)		
} else {		
motion_vertical_field_select [0][s]	1	uimsbf
motion_vector(0,s)		
motion_vertical_field_select [1][s]	1	uimsbf
motion_vector(1,s)		
}		

【図 64】

motion_vectors(r,s) {	No. of bits	Mnemonic
motion_code[r][s][0]	1-11	viclbf
if ((f_code[s][0]!=1)&&(motion_code[r][s][0]!=0))		
motion_residual [r][s][0]	1-8	uimsbf
if (dmv==1)		
dmvector [0]	1-2	viclbf
motion_code[r][s][1]	1-11	viclbf
if ((f_code[s][1]!=1)&&(motion_code[r][s][1]!=0))		
motion_residual [r][s][1]	1-8	uimsbf
if (dmv==1)		
dmvector [1]	1-2	viclbf
}		

【図 65】

macroblock_type V L C code					
				macroblock_quant	
				dct_type_flag	
				macroblock_motion_forward	
				macroblock_motion_backward	
				Description	
1	0	1	0	0	Intra
01	1	1	0	0	Intra, Quant

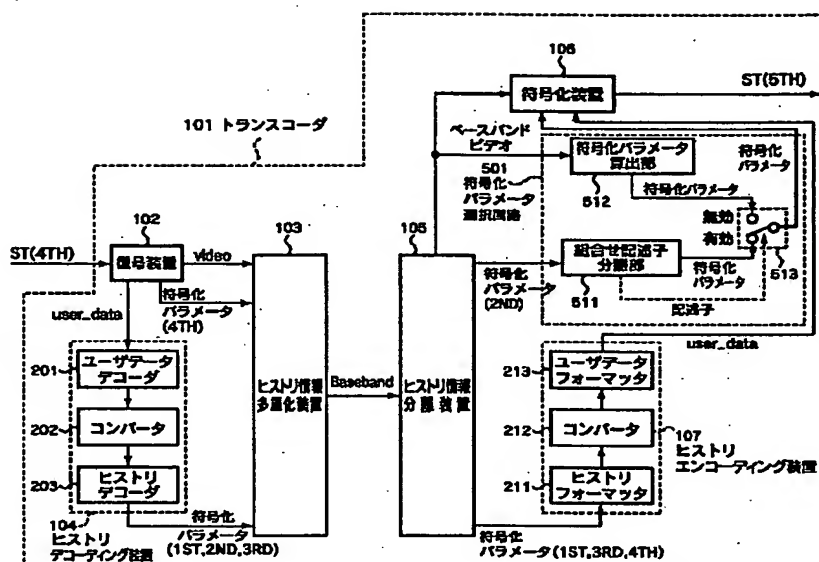
【図 66】

macroblock_type V L C code					
		macroblock_quant			
		dct_type_flag			
		macroblock_motion_forward			
		macroblock_motion_backward			
		Description			
1	0	1	1	0	MC, Coded
01	0	1	0	0	No MC, Coded
001	0	0	0	0	MC, Not Coded
0001 1	0	1	0	0	Intra
0001 0	1	1	1	0	MC, Coded, Quant
0000 1	1	1	0	0	No MC, Coded, Quant
0000 01	1	1	0	0	Intra, Quant

【图 6 7】

macroblock_type V L C code					macroblock_quant	dict_type_flag	macroblock_motion_forward	macroblock_motion_backward	Description
10	0	0	0	0					Interp, Not Coded
11	0	1	1	1					Interp, Coded
010	0	0	0	0					Bwd, Not Coded
011	0	1	0	1					Bwd, Coded
0010	0	0	0	0					Fwd, Not Coded
0011	0	1	1	0					Fwd, Coded
0001 1	0	1	0	0					Intra
0001 0	1	1	1	1					Interp, Coded, Quant
0000 11	1	1	1	0					Fwd, Coded, Quant
0000 10	1	1	0	1					Bwd, Coded, Quant
0000 01	1	1	0	0					Intra, Quant

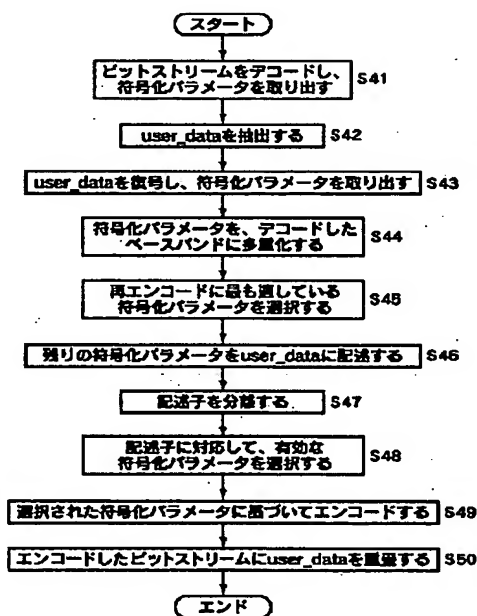
【图 68】



【图 6 9】

[illegible]

【图 70】



【图 7-2】

項目の組合せ	red_bw_flag	
	red_bw_indicator	
組合せ1	0	—
組合せ2	1	0
組合せ3	1	1
組合せ4	1	2
組合せ5	1	3

【図 7 1】

re_coding_stream_info () {	No. of bits	Mnemonic
user_data_start_code	32	bslbf
re_coding_stream_info_ID	16	bslbf
red_bw_flag	1	ulmsbf
if (red_bw_flag) {		
red_bw_indicator	2	ulmsbf
if (ired_bw_flag) {		
for(i=0;i<number_of_macroblock++;) {		
marker_bit	3	bslbf
num_other_bits	7	ulmsbf
num_mv_bits	8	ulmsbf
num_coef_bits	14	ulmsbf
}		
}		
next_start_code()		
}		

【図 7 3】

Line	Frame Coding	Field Coding	
0	0	0	SRIB_sync_code=111112 Reserved: 0 fr fl SRIB SRIB_top SRIB_repeat 422 420 chroma 0
		1	rolling_SRIB_mb_ref[15:0]
1	2		picrate_element[picrate_element_index][31:16]
		3	picrate_element[picrate_element_index][15:0]
2	4		mb_quant mb_intra mb_slice mv[0][0][0][12:0]
		5	mb_pattern mb_intra mv_vert_field_sel[r][s] mv[0][0][1][8:0]
3	6		DCT_type motion_type mv[0][1][0][12:0]
		7	skip_mb q_scale_code[4:0] mv[0][1][1][8:0]
4	8		Reserved mv[1][0][0][12:0]
		9	coded_block_pattern[7:0] Reserved mv[1][0][1][8:0]
5	10		Reserved mv[1][1][0][12:0]
		11	coded_block_pattern[3:0] Reserved mv[1][1][1][8:0]
6	12		Reserved num_other_bits[6:0] num_mv_bits[7:2]
		13	num_mv_bits[1:0] num_coef_bits[13:0]
7	14		SRIB_crc[31:16]
		15	SRIB_crc[15:0]

【図 7 4】

picture rate elements(74-1)

parameter	number format	number of bits	bit offset from	bit offset to	data category	details
MPEG standard flag	1bit flag	1	0	0	3	1=>MPEG1.0=>MPEG2
red_bw_flag	1bit flag	1	1	1	3	Default="0"
red_bw_indicator	2bit ui	3	2	4	3	Default="000"
header present flags	2bit flags	2	5	6	3	sequence header present flag, GOP header present flag
Extension start code flags	16 flags	16	7	22	3	Indicates if a given extension start code exists. The 16 flags correspond to the 16 entries in table 6.2 of the ISO/IEC 13818-2: 1998 standard in the order they are listed.
Other start codes	3 flags	3	23	25	3	user_data_start_code, sequence_error_code, sequence_end_code
sequence header						
horizontal_size	14bit uimsbf	14	26	39	2	includes extension
vertical_size	14bit uimsbf	14	40	53	2	includes extension
aspect_ratio_information	4bit uimsbf	4	54	57	2	
frame_rate_code	4bit uimsbf	4	58	61	2	
bit_rate	30bit uimsbf	30	62	91	2	includes extension, correct value should be calculated
vbv_buffer_size	18bit uimsbf	18	92	109	2	includes extension
constrained_parameters_flag	1bit flag	1	110	110	2	
sequence extension						
profile_and_level_indication	8bit uimsbf	8	111	118	2	
progressive_sequence	1bit flag	1	119	119	2	
chroma_format	2bit uimsbf	2	120	121	2	
low_delay	1bit flag	1	122	122	2	
sequence display extension						
video_format	3bit uimsbf	3	123	125	2	
colour_description	1bit flag	1	126	126	2	
colour_primaries	8bit uimsbf	8	127	134	2	
transfer_characteristics	8bit uimsbf	8	135	142	2	
matrix_coefficients	8bit uimsbf	8	143	150	2	
display_horizontal_size	14bit uimsbf	14	151	164	2	
display_vertical_size	14bit uimsbf	14	165	178	2	
group of pictures header						
time_code	25bit flag	25	179	203	2	
closed_gop	1bit flag	1	204	204	2	

【図 7 5】

picture rate elements(74-2)

broken_link	1bit flag	1	205	205	2	
picture_header						
temporal_reference	10bit uimsbf	10	206	215	1	
picture_coding_type	3bit uimsbf	3	216	218	1	
vbv_delay	16bit uimsbf	16	219	234	1	should be calculated if not present in bitstream
full_pel_forward_vector	1bit flag	1	235	235	1	
forward_f_code	3bit uimsbf	3	236	238	1	
full_pel_backward_vector	1bit flag	1	239	239	1	
backward_f_code	3bit uimsbf	3	240	242	1	
picture_coding_extension						
forward_horizontal_f_code	4bit uimsbf	4	243	246	1	
forward_vertical_f_code	4bit uimsbf	4	247	250	1	
backward_horizontal_f_code	4bit uimsbf	4	251	254	1	
backward_vertical_f_code	4bit uimsbf	4	255	258	1	
Intra_dc_precision	2bit uimsbf	2	259	260	1	
picture_structure	2bit uimsbf	2	261	262	1	
top_field_first	1bit flag	1	263	263	1	
frame_pred_frame_dct	1bit flag	1	264	264	1	
concealment_motion_vectors	1bit flag	1	265	265	1	
q_scale_type	1bit flag	1	266	266	1	
Intra_vlc_format	1bit flag	1	267	267	1	
alternate_scan	1bit flag	1	268	268	1	
repeat_first_field	1bit flag	1	269	269	1	
chroma_420_type	1bit flag	1	270	270	1	
progressive_frame	1bit flag	1	271	271	1	
composite_display_flag	1bit flag	1	272	272	1	
v_axis	1bit flag	1	273	273	1	
field_sequence	3bit uimsbf	3	274	276	1	
sub_carrier	1bit flag	1	277	277	1	
burst_amplitude	7bit uimsbf	7	278	284	1	
sub_carrier_phase	8bit uimsbf	8	285	292	1	
quant_matrix_extension						
load_intra_quantiser_matrix	1bit flag	1	293	293	1	(1)
load_non_intra_quantiser_matrix	1bit flag	1	294	294	1	(1)
load_chroma_intra_quantiser_matrix	1bit flag	1	295	295	1	(1)
load_chroma_non_intra_quantiser_matrix	1bit flag	1	296	296	1	(1)
intra_quantiser_matrix[64]	64*0...255	512	297	808	2	(1)
non_intra_quantiser_matrix[64]	64*0...255	512	809	1320	2	(1)
chroma_intra_quantiser_matrix[64]	64*0...255	512	1321	1832	2	(1)
chroma_non_intra_quantiser_matrix[64]	64*0...255	512	1833	2344	2	(1)
picture_display_extension						
frame_centre_horizontal_offset_1	16bit uimsbf	16	2345	2360	2	
frame_centre_vertical_offset_1	16bit uimsbf	16	2361	2376	2	
frame_centre_horizontal_offset_2	16bit uimsbf	16	2377	2392	2	

【図 7 6】

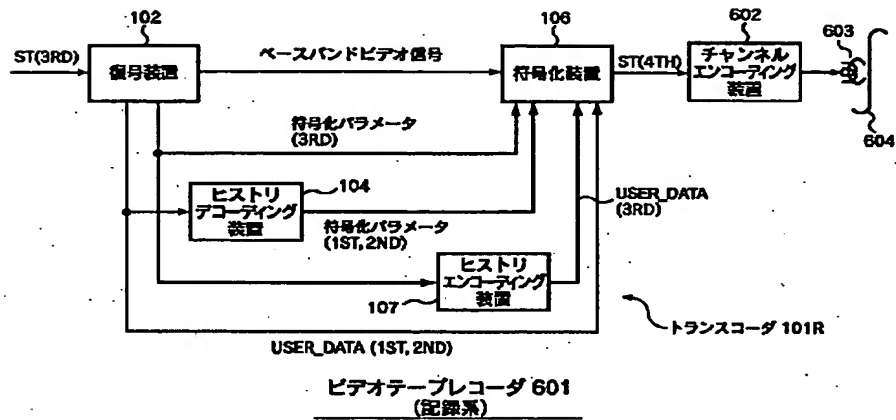
picture rate elements(74-3)

frame_centre_vertical_offset_2	16bit uimsbf	16	2393	2408	2	
frame_centre_horizontal_offset_3	16bit uimsbf	16	2409	2424	2	
frame_centre_vertical_offset_3	16bit uimsbf	16	2425	2440	2	
copyright extension						
copyright_flag	1bit flag	1	2441	2441	2	
copyright_identifier	8bit code	8	2442	2449	2	
original_or_copy	1bit flag	1	2450	2450	2	
copyright_number	64bit uimsbf	64	2451	2514	2	
PTS/DTS						
PTS_DTS_flag	2bit flag	2	2515	2516	1	
PTS_Value	33bit uimsbf	33	2517	2549	2	
DTS_Value	33bit uimsbf	33	2550	2582	2	
spare reserved bits						
spare	41bit uimsbf	41	2583	2623		
user data area						
user_data		1664	2624	4287	2	
picture rate information CRC						
32-bit_protection_CRT	32bit uimsbf	32	4288	4319		

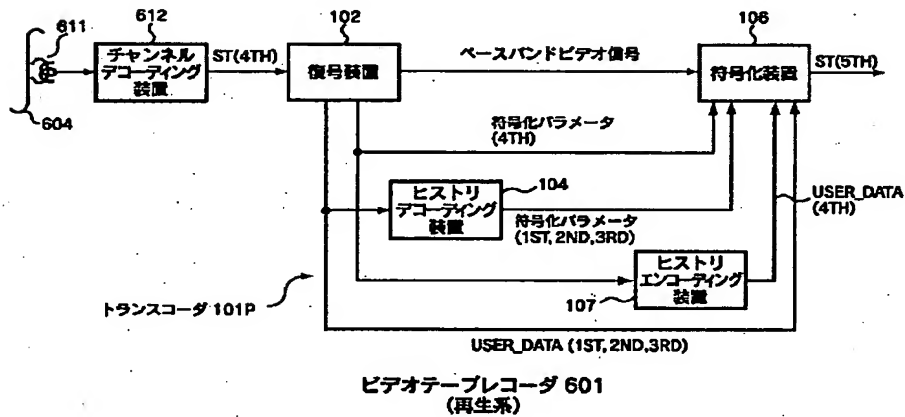
【図 7 7】

D9	Cb[0][9]	Y[0][9]	Cr[0][9]	Y[1][9]	Cb[1][9]	Y[2][9]	Cr[1][9]	Y[3][9]
D8	Cb[0][8]	Y[0][8]	Cr[0][8]	Y[1][8]	Cb[1][8]	Y[2][8]	Cr[1][8]	Y[3][8]
D7	Cb[0][7]	Y[0][7]	Cr[0][7]	Y[1][7]	Cb[1][7]	Y[2][7]	Cr[1][7]	Y[3][7]
D6	Cb[0][6]	Y[0][6]	Cr[0][6]	Y[1][6]	Cb[1][6]	Y[2][6]	Cr[1][6]	Y[3][6]
D5	Cb[0][5]	Y[0][5]	Cr[0][5]	Y[1][5]	Cb[1][5]	Y[2][5]	Cr[1][5]	Y[3][5]
D4	Cb[0][4]	Y[0][4]	Cr[0][4]	Y[1][4]	Cb[1][4]	Y[2][4]	Cr[1][4]	Y[3][4]
D3	Cb[0][3]	Y[0][3]	Cr[0][3]	Y[1][3]	Cb[1][3]	Y[2][3]	Cr[1][3]	Y[3][3]
D2	Cb[0][2]	Y[0][2]	Cr[0][2]	Y[1][2]	Cb[1][2]	Y[2][2]	Cr[1][2]	Y[3][2]
D1	Cb[0][1]	Y[0][1]	Cr[0][1]	Y[1][1]	Cb[1][1]	Y[2][1]	Cr[1][1]	Y[3][1]
D0	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[0][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[1][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[2][0]	Embedded Aligned MPEG-2 Re-Coding Information Bus	Y[3][0]

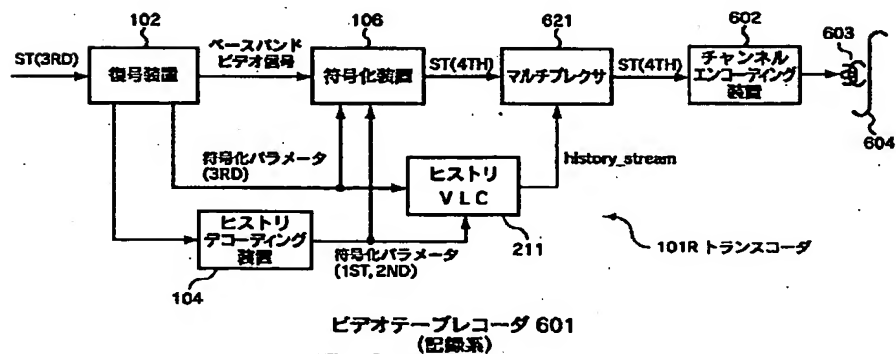
【図 78】



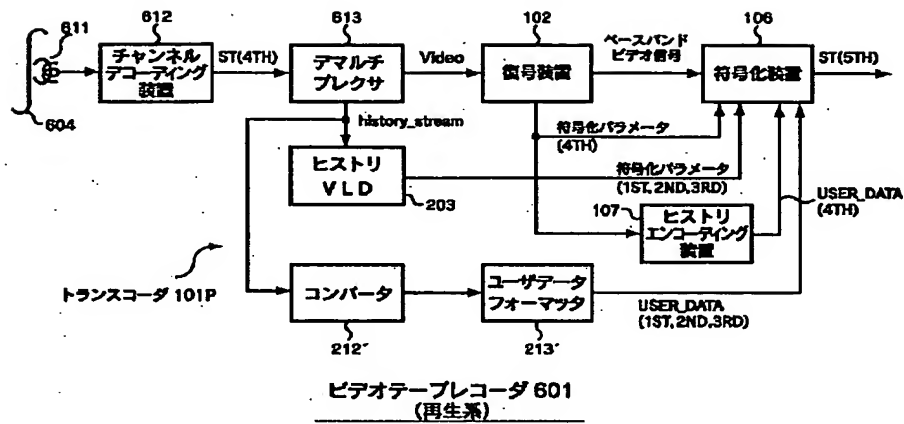
【図 79】



【図 80】



【図 81】



【手続補正書】

【提出日】平成11年12月10日(1999.12.

【補正対象項目名】図3

10)

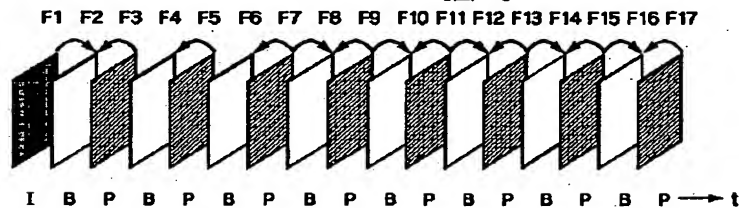
【補正方法】変更

【手続補正1】

【補正内容】

【補正対象書類名】図面

【図3】



B-ピクチャ (両方向予測)

ピクチャタイプ I,P,B-picture

【手続補正2】

【補正方法】変更

【補正対象書類名】図面

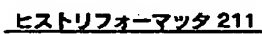
【補正内容】

【補正対象項目名】図12

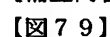
【図12】



【図 20】



【图 3 9】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.